Algoritmica Avanzata – Compito, 20/6/2005 (Durata: 3h)

Studente (Nome, Cognome): _____

Matricola: _____

Prima Parte: domande di teoria

Si forniscano le risposte alle seguenti due domande teoriche:

1. Dimostrare che $\langle G = (V, E), k\rangle \in$ CLIQUE $\Leftrightarrow \langle G^c = (V, E^c), k\rangle \in$ INDEPENDENT SET.

   Solution:

   $$\begin{aligned}
   &\langle G = (V, E), k\rangle \in \text{CLIQUE} \\
   &\quad \Leftrightarrow \quad \exists V' \subseteq V, |V'| = k : \forall u, v \in V, u \neq v : u, v \in V' \Rightarrow \{u, v\} \in E \\
   &\quad \Leftrightarrow \quad \exists V' \subseteq V, |V'| = k : \forall u, v \in V, u \neq v : u, v \in V' \Rightarrow \{u, v\} \notin E^c \\
   &\langle G^c = (V, E^c), k\rangle \in \text{INDEPENDENT SET}
   \end{aligned}$$

2. Siano $P = (e, n)$ e $S = (d, n)$ una coppia di chiavi RSA. Si dimostri che:
   $$\forall M \in \boldsymbol{Z}_n : M = S(P(M)) = P(S(M)).$$

   Solution:

   Let $n = pq$, with $p \neq q$ primes, and consider and arbitrary message $M \in \boldsymbol{Z}_n$. Since $P(M) = M^e \bmod n$ and $S(M) = M^d \bmod n$, we have to show that $P(S(M)) = S(P(M)) = M^{ed} \bmod n = M$. We will prove that $M \equiv M^{ed} \bmod p$ and $M \equiv M^{ed} \bmod q$, whence $M^{ed} \equiv M \bmod n$ from the Chinese Reminder Theorem, which in turn implies that $M^{ed} \bmod n = M$

   Consider p first. If $M \equiv 0 \bmod p$, then $M^{ed} \equiv 0 \bmod p$. Hence $M^{ed} \equiv M \bmod p$. Otherwise ($M \not\equiv 0 \bmod p$) it must be $\gcd(M, p) = 1$ (since p is prime). Since $ed = 1 + k\phi(n)$, for some $k \in \boldsymbol{Z}$, we have $M^{ed} \bmod p = M^{1+k\phi(n)} \bmod p = M^{1+k(p-1)(q-1)} \bmod p = \left((M \bmod p) \cdot (M^{p-1} \bmod p)^{k(q-1)}\right) \bmod p = M \bmod p$ by Fermat's Theorem.

   The proof that $M^{ed} \equiv M \bmod q$ is identical.

Seconda Parte: risoluzione di problemi

Esercizio 1 [12 punti] Dato un grafo non orientato $G = (V, E)$, un *dominating set* $D \subseteq V$ di G è un sottoinsieme di vertici tale che, per ogni nodo non isolato $v \in V$ (ovvero, ogni nodo v su cui incide almeno un arco), D contiene v oppure un suo vicino. Formalmente:

$$\forall v \in V : v \text{ non isolato} \Rightarrow (v \in D) \vee (\exists \{u, v\} \in E : u \in D).$$

Si consideri il seguente problema decisionale:

DOMINATING SET:
ISTANZA: $\langle G = (V, E), k \rangle$, con $k \leq |V|$
DOMANDA: Esiste un dominating set $D \subseteq V$ con $|D| = k$?

Si vuole dimostrare che DOMINATING SET è NP-Hard. Si consideri la seguente riduzione f da VERTEX COVER: data l'istanza $\langle G = (V, E), h \rangle$, per ogni arco $\{u, v\} \in E$ si aggiunga al grafo un nuovo nodo z_{uv} e due nuovi archi che connettono z_{uv} a u e v. Si ponga infine $k = h$.

1. Definire formalmente f e dimostrare che essa è calcolabile in tempo polinomiale.
2. Si dimostri: $x \in$ VERTEX COVER $\Rightarrow f(x) \in$ DOMINATING SET.
3. Si dimostri: $f(x) \in$ DOMINATING SET $\Rightarrow x \in$ VERTEX COVER.

Answer:

(**Point 1**) The suggested reduction is $f(\langle G = (V, E), k \rangle) = \langle G' = (V', E'), h \rangle$, where

1. $V' = V \cup \{z_{uv} : \{u, v\} \in E\}$. (For each edge $\{u, v\} \in E$ we add a new node z_{uv} to $V' \ldots$)
2. $E' = E \cup \{\{u, z_{uv}\}, \{v, z_{uv}\} : \{u, v\} \in E\}$. ($\ldots$ and connect $z_{u,v}$ to u and v only.)
3. $h = k$ (We look for a Dominating Set of G' as large as the Vertex Cover of G).

G' has $|V|+|E|$ nodes and $3|E|$ edges, hence f can be computed in linear time in $|\langle G = (V, E), k \rangle|$.
(**Point 2**) If $x = \langle G = (V, E), k \rangle \in$ VERTEX COVER, there exists a subset $\widehat{V} \subseteq V$ of size k such that each edge $\{u, v\} \in E$ has at least one endpoint in $\widehat{V}$. We now show that $\widehat{V} \subseteq V'$ is also a dominating set of G'. Consider any non-isolated node $y \in V'$. If $y \in V' \cap V$, then there exists an edge $\{u, y\} \in E$. Since $\widehat{V}$ is a vertex-cover, either $y \in \widehat{V}$ or $u \in \widehat{V}$. If $y \notin V' \cap V$, then there exists an edge $\{u, v\} \in E$ such that $y = z_{u,v}$, and y has u and v as neighbors. Again, one of these latter two nodes must be in $\widehat{V}$, since $\{u, v\} \in E$. The two cases together show that the dominating set condition holds for all non-isolated nodes, hence $\widehat{V}$ is a dominating set in G' of size k. Therefore $f(x) = \langle G' = (V', E'), k \rangle \in$ DOMINATING SET.
(**Point 3**)If $f(x) = \langle G' = (V', E'), k \rangle \in$ DOMINATING SET, there exists a dominating set $D \subseteq V'$ of size k. Observe that D may contain nodes $z_{u,v} \notin V$, so we cannot claim directly that D is a vertex cover in G. However, we may substitute each node $z_{u,v} \in D$ with one of the two

nodes u or v and still obtain a dominating set $D' \subseteq V$ of size at most k, since $z_{u,v}$ can only be used to dominate itself, u or v, and the same task can be accomplished by either u or v. Consider now an arbitrary $\{u, v\} \in E$. D' must contain either u or v or otherwise the domination condition would not hold for $z_{u,v} \notin D'$. Therefore D' is a vertex cover for G of size $\leq k$, which in turn implies that there is a vertex cover in G of size k (which can be obtained by adding $k - |D'|$ arbitrary nodes in V to D'). Therefore $x = \langle G = (V, E), k\rangle \in$ VERTEX COVER. □

Esercizio 2 [10 punti] Sia $\{X_i\}_{1\leq i\leq n}$ un insieme di n variabili indicatore indipendenti con $\Pr(X_i = 1) = 1/(4e)$. Sia $X = \sum_{i=1}^{n} X_i$. Usando un bound di Chernoff, si dimostri che

$$\Pr(X > n/2) < \frac{1}{(\sqrt{2})^n}.$$

Answer: Let $p = 1/(4e)$ and $\mu = E[X] = np = n/(4e)$. Let us write $n/2$ as $(1 + \delta)\mu$ so that we can apply the following Chernoff bound:

$$\Pr(X > (1 + \delta)\mu) < \left(\frac{e^\delta}{(1 + \delta)^{1+\delta}}\right)^\mu$$

We have $n/2 = (1 + \delta)n/(4e) \Leftrightarrow \delta = 2e - 1$. Therefore

$$\begin{aligned}
\Pr(X > n/2) &= \Pr(X > (1 + (2e - 1))\mu) \\
&< \left(e^{2e-1}/(2e)^{2e})\right)^{n/(4e)} \\
&< 1/(2^{2e/4e})^n \\
&= \left(1/\sqrt{2}\right)^n
\end{aligned}$$

□

Esercizio 3 [11 punti] Siano dati un poligono convesso $P = \{p_0, p_1, \ldots, p_{n-1}\}$, con i vertici ordinati in senso antiorario, e un punto q esterno a P e non collineare ad alcun lato di P. Un *punto di tangenza su P da q* è un vertice $p_i \in P$ tale che la retta che passa per q e p_i interseca il perimetro di P esclusivamente in p_i. Si dia lo pseudocodice e si analizzi un algoritmo TANGENT(P, q) che ritorna un punto di tangenza su P da q in tempo $O(n)$.
(*Suggerimento*: Per ogni vertice p_i, si confrontino le svolte di due opportune spezzate che partono da $q \ldots$)

Answer: The algorithm is based on the following simple observation: *an arbitrary vertex $p_i \in P$ is a tangent point on P from q if and only if $\overrightarrow{qp_ip_{i-1}}$ and $\overrightarrow{qp_ip_{i+1}}$ (where arithmetic on the indices is performed modulo n) turn in the same direction.* To prove this, first observe that if p_i is a tangent point on P from q, then P lies entirely in one subplane generated by the straight line r containing $\overline{qp_i}$, hence $\overrightarrow{qp_ip_{i-1}}$ and $\overrightarrow{qp_ip_{i+1}}$ must turn in the same direction. Vice versa, if the two turns are concordant, then sides $\overline{p_ip_{i-1}}$ and $\overline{p_ip_{i+1}}$ are in the same subplane generated by r. However, the polygon is convex, hence P must lie completely in that subplane. Since q is not collinear with any side of P, every other point on the perimeter different from p_i is strictly at the left or at the right of r, hence no other intersection is possible.

For each p_i, the test can be easily implemented in constant time by comparing the signs of the cross-products $(p_i - q) \times (p_{i-1} - q)$ and $(p_i - q) \times (p_{i+1} - q)$.

The pseudocode of the algorithm follows.

```
TANGENT(P, q)
    * Let P = {p_0, p_1, ..., p_{n-1}} *
    p_{-1} ← p_{n-1}   p_n ← p_0
    {... so we won't have to use modular
         arithmetic on the indices ...}
    for i ← 0 to n − 1 do
        cp_1 ← (p_i − q) × (p_{i-1} − q)
        cp_2 ← (p_i − q) × (p_{i+1} − q)
        if (cp_1 · cp_2 > 0)
            then return p_i
```

Observe that if q is not collinear with a side of p, then there must surely exist a tangent point on P from q (indeed, there are exactly two such points) and the algorithm will correctly return at some iteration of the **for** loop.

The correctness of the algorithm follows from the above discussion. As for its running time, it is clearly linear in the number of vertices of the polygon. □

Algoritmica Avanzata – Compito, 4/7/2005 (Durata: 3h)

Studente (Nome, Cognome): _____

Matricola: _____

Prima Parte: domande di teoria

Si forniscano risposte **concise** e **rigorose** alle seguenti due domande teoriche:

1. Dimostrare che se $P \neq NP$, per ogni ρ intero costante il problema di ottimizzazione associato al *Traveling Salesman Problem (TSP)* non è ρ-approssimabile in tempo polinomiale.

   Solution: In order to show that there cannot exist a polynomial ρ-approximation algorithm with $\rho = O(1)$ for the optimization version of TSP (call it *opt-TSP*) under the hypothesis $P \neq NP$, we show that any such algorithm would imply that HAMILTON $\in P$, which in turn implies $P = NP$, a contradiction.

   Consider any instance $G = \langle V, E \rangle$ of HAMILTON. In polynomial time, we can obtain the following instance of opt-TSP:

   (a) $G' = (V, E')$, the complete graph built on the set of nodes V;

   (b) Cost function $c : V \times V \rightarrow \boldsymbol{Z}$, with $c(u, v) = c(v, u) = \begin{cases} 1 & \{u, v\} \in E \\ \rho|V| + 1 & \{u, v\} \in E' - E. \end{cases}$

   If $G = \langle V, E \rangle \in$ HAMILTON then there is a hamiltonian circuit in G, hence, by construction, the optimal solution of the corresponding instance of opt-TSP has cost $|V|$, and the approximation algorithm must return a solution with a cost not greater than $\rho|V|$. Vice versa, if $G = \langle V, E \rangle \notin$ HAMILTON any tour in G' must use an edge in $E' - E$, hence its cost is at least $|V| - 1 + \rho|V| + 1 \geq \rho|V| + 1$. Therefore it is sufficient to run the hypothetical polynomial approximation algorithm and check (in linear time) the cost of the returned solution to decide HAMILTON in polynomial time.

2. Dimostrare che se un multigrafo connesso $\mathcal{G} = (V, E)$ ha un taglio minimo di cardinalità k, allora $|E| \geq k|V|/2$.

   Solution: Consider an arbitrary node $v \in V$, and let $\deg(v)$ denote its degree. Since the multiset of edges $\{\{u, v\} \in E\}$ is a cut of size $\deg(v)$ disconnecting $\{v\}$ from $V - \{v\}$, and k is the size of the minimum cut, it must be $\deg(v) \geq k$. But then

   $$|E| = \sum_{v \in V} \deg(v)/2 \geq k|V|/2.$$

Seconda Parte: risoluzione di problemi

Esercizio 1 [12 punti] Si consideri il seguente problema decisionale:

PARTITION:

ISTANZA: $\langle S \rangle$, $S \subset \boldsymbol{N}$ insieme finito

DOMANDA: Esistono $S_1, S_2 \subset S$, con $S_1 \cup S_2 = S$ e $S_1 \cap S_2 = \emptyset$ tali che $\sum_{s \in S_1} s = \sum_{s \in S_2} s$?

Si dimostri che PARTITION è NP-Hard.

(*Suggerimento:* Si riduca da SUBSET SUM (SS) trattando separatamente casi semplici e, per il caso generale, aggiungendo all'insieme S dell'istanza di SS due numeri appropriati ...)

Answer: As suggested in the text, we reduce from SS. Let $\langle S, t \rangle$ be an instance of SS, and define $M = \sum_{s \in S} s$. We consider two "easy" cases first. If $t > M$, then the instance is clearly a negative one, and we can map it to any negative instance of PARTITION, (say $\langle \{1\} \rangle$). If $t = M/2$, then there is a subset of S summing to t if and only if S can be partitioned into two disjoint sets of size t each, hence we can simply map $\langle S, t \rangle$ to $\langle S \rangle$.

We are left to deal with instances with $M/2 \neq t \leq M$. Note that if any such instance $\langle S, t \rangle \in$ SS, then S can be partitioned into two subsets of sum t and $M - t$. The idea is to build on this partition, making sure that in the solution of the reduced instance, these two subsets end up one in S_1 and the other in S_2. To this purpose, we add two "large" numbers $2M - t + 1$ and $M + t + 1$, observing that both are greater than M, hence they cannot belong to S, and are distinct when $t \neq M/2$. In summary, our reduction function is the following:

$$f(\langle S, t \rangle) = \begin{cases} \langle S \cup \{2M - t + 1, M + t + 1\} \rangle, & M = \sum_{s \in S} s \text{ and } M/2 \neq t \leq M; \\ \langle S \rangle & t = M/2; \\ \langle \{1\} \rangle & t > M. \end{cases}$$

Function f is clearly computable in polynomial time, since adding up all numbers in S cannot take more than quadratic time in $|\langle S \rangle|$ (a finer argument shows that the transformation is linear).

Let us now show that f is indeed a reduction from SS to PARTITION. If $t = M/2$ or $t > M$ we have already argued that the reduction is correct. Consider now the case $M/2 \neq t \leq M$. If $\langle S, t \rangle \in$ SS then there exists a subset $S' \subseteq S$ such that $\sum_{s \in S'} s = t$. But then sets $S_1 = S' \cup \{2M - t + 1\}$ and $S_2 = (S - S') \cup \{M + t + 1\}$ form a partition of $S \cup \{2M - t + 1, M + t + 1\}$ into two disjoint sets with $\sum_{s \in S_1} s = \sum_{s \in S_2} s = 2M + 1$, hence $f(\langle S, t \rangle) \in$ PARTITION.

Vice versa, assume that $f(\langle S, t \rangle) \in$ PARTITION. Since $\sum_{s \in S \cup \{2M - t + 1, M + t + 1\}} s = 4M + 2$, the two sets S_1 and S_2 must sum to $2M + 1$ each. However, all elements of S sum to M, therefore each of S_1 and S_2 must contain exactly one of the two added elements. Without loss of generality, assume that $2M - t + 1 \in S_1$. Then $S' = S_1 - \{2M - t + 1\}$ is such that $S' \subseteq S$ and $\sum_{s \in S'} s = 2M + 1 - (2M - t + 1) = t$, hence $\langle S, t \rangle \in$ SS. $\square$

Esercizio 2 [11 punti] Si considerino istanze non vuote $\langle S, t\rangle$ del problema di ottimo relativo a SUBSET SUM tali che $\forall s \in S : s \leq t$. Si consideri il seguente algoritmo che prende in ingresso una di tali istanze:

```
APPROX_OPT_SS(⟨S, t⟩)
{s_1, s_2, ..., s_n} ← SORT-DECREASING(S)
sum ← s_1
for i ← 2 to n do
    if (sum + s_i ≤ t)
        then sum ← sum + s_i
        else return sum
return sum
```

Nel codice, SORT-DECREASING(S) rappresenta una chiamata a una routine di ordinamento che ritorna gli elementi dell'insieme S in ordine **decrescente**. Si dimostri che APPROX_OPT_SS è un algoritmo di 2-approssimazione per le istanze di SS definite sopra.

Answer: Since sum is initialized to $s_1 \leq t$ and an element $s_i \in S$ is added to sum only if $sum + s_i \leq t$, the value returned is indeed the cost of a feasible solution. Let $s^\star$ be the optimal cost for the instance. It suffices to show that the value sum returned is such that $s^\star/sum \leq 2$, which immediately implies that APPROX_OPT_SS is a 2-approximation algorithm.

Consider first the case when the algorithm returns from outside the the **for** loop. Then on exit $sum = \sum_{s\in S} s$, which clearly implies that $sum = s^\star$, whence $s^\star/sum = 1 \leq 2$. Assume now that the algorithm returns from within the **for** loop. Then, there is an index i', $2 \leq i' \leq n$ such that $sum + s_{i'} > t$. However, $s_{i'} < s_1$ by the sorting, and $s_1 \leq sum$ since s_1 is always added to sum. Therefore we obtain $2 \cdot sum > sum + s_{i'} > t$ which implies $sum > t/2$. Hence $s^\star/sum < t/(t/2) = 2$. □

Esercizio 3 [10 punti] Si consideri un criptosistema a chiave pubblica di tipo RSA in cui il dominio dei messaggi è $\boldsymbol{Z}_n$, con $n = 7 \times 19 = 133$. Data la chiave pubblica $P = (7, 133)$ determinare la corrispondente chiave segreta S e la relativa funzione di codifica $S(M)$ ad essa associata.

Answer: If $n = 7 \times 19 = 133$, we have that $\phi(n) = 6 \times 18 = 108$. The given public key is $P = (7, 133)$, whose associated encryption function is $P(M) = M^7 \bmod 133$, for each $M \in \boldsymbol{Z}_{133}$. Under RSA, the corresponding secret key will be (d, n), where d is the multiplicative inverse of 7 in $\boldsymbol{Z}^\star_{108}$. The value of d can be inferred by writing 1=gcd(108,7) as an integer linear combination of 108 and 7 (Bezout's equality). The coefficients of such combination are obtained by applying EXTENDED_EUCLID (EE) to 108 and 7. We obtain the following recursive calls:

EE(108,7) → EE(7,3) → EE(3,1) → EE(1,0)

whose return values (in inverse order) are:

$\{1, (1, 0)\} \rightarrow \{1, (0, 1)\} \rightarrow \{1, (1, 0 - 2 \cdot 1)\} \rightarrow \{1, (-2, 1 - (15) \cdot (-2))\} = \{1, (-2, 31)\}$

Since $31 \cdot 7 = 1 + 2 \cdot 108$, we have that in $\boldsymbol{Z}^\star_{108}$, $d = e^{-1} = 31$, hence $S = (31, 133)$ and $S(M) = (M^{31} \bmod 133)$ for each $M \in \boldsymbol{Z}_{133}$. □

Algoritmica Avanzata – Compito, 8/9/2005 (Durata: 3h)

Studente (Nome, Cognome): _____

Matricola: _____

Prima Parte: domande di teoria

Si forniscano le risposte alle seguenti due domande teoriche:

1. Dimostrare che $\langle G = (V, E), k\rangle \in$ CLIQUE $\Leftrightarrow \langle G^c = (V, E^c), |V|-k\rangle \in$ VERTEX COVER.

   Solution:

   $$\begin{aligned}
   &\langle G = (V, E), k\rangle \in \text{CLIQUE} \\
   &\quad\Leftrightarrow\ \exists V' \subseteq V, |V'| = k : \forall u, v \in V, u \neq v : [(u \in V') \wedge (v \in V')] \Rightarrow \{u, v\} \in E \\
   &\quad\Leftrightarrow\ \exists V' \subseteq V, |V'| = k : \forall u, v \in V, u \neq v : \{u, v\} \notin E \Rightarrow [(u \notin V') \vee (v \notin V')],
   \end{aligned}$$

   where the last proposition is simply obtained by reversing the implication using its equivalent counter-positive formulation. Let us now rewrite this proposition by introducing the new quantified variable $V'' = V - V'$ and observing that $|V''| = |V| - |V'|$, $\{u, v\} \notin E \Leftrightarrow \{u, v\} \in E^c$ and $(u \notin V') \Leftrightarrow (u \in V'')$:

   $$\Leftrightarrow\ \exists V'' \subseteq V, |V''| = |V| - k : \forall u, v \in V, u \neq v : \{u, v\} \in E^c \Rightarrow [(u \in V'') \vee (v \in V'')] \Leftrightarrow$$

   $\langle G^c = (V, E^c), |V| - k\rangle \in$ VERTEX COVER

2. Dato un grafo non orientato $G = (V, E)$, sia $A \subseteq E$ tale che:
   1) $\forall e_1 \neq e_2 \in A : e_1 \cap e_2 = \emptyset$ (cioè, e_1 e e_2 non hanno estremi in comune)
   2) $\forall e' \in E - A\ \exists e'' \in A : e' \cap e'' \neq \emptyset$. (cioè, e' e e'' condividono un estremo)

   Si dimostri che un minimo vertex cover $V^\star$ di G soddisfa: $|A| \leq |V^\star| \leq 2|A|$.

   Solution:

   To show that $|V^*| \leq 2|A|$, it is sufficient to observe that the set of all vertices V_A of edges in A, whose size is $2|A|$, is a vertex cover, since V_A trivially covers the edges in A while Property 2 implies that each edge in $E - A$ has one endpoint in V_A. To show that $|V^*| \geq |A|$, observe that by Property 1 all edges in A are disjoint, hence at least one endpoint for each such edge must be present in every vertex cover.

Seconda Parte: risoluzione di problemi

Esercizio 1 [10 punti] Siano L_1 e L_2 due linguaggi NP-Hard e si supponga che esista una funzione di riduzione f da SAT a L_1 calcolabile in tempo polinomiale tale che, per ogni stringa $x \in \{0,1\}^\star$, si abbia inoltre che $f(x) \notin L_2$. Si dimostri che allora il linguaggio $L_1 \cup L_2$ è anch'esso NP-Hard.

Answer: It is sufficient to prove that under the stated hypotheses, f also reduces SAT to $L_1 \cup L_2$. Indeed, if $x \in$ SAT, then $f(x) \in L_1$, since f reduces SAT to L_1, hence $f(x) \in L_1 \cup L_2$. Vice versa, if $f(x) \in L_1 \cup L_2$, it must be that $f(x) \in L_1$ (since $f(x) \notin L_2$ from the hypothesis), hence $x \in$ SAT (again, since f reduces SAT to L_1). Observe that the hypothesis $L_2 \in$ NPH is not used in the above argument, hence the result holds regardless of the complexity of accepting language L_2. □

Esercizio 2 [11 punti] Date n scatole e $m \geq 16n \log_e n$ palline, si supponga di porre ogni pallina in una scatola in modo casuale. Si dimostri che nell'allocazione risultante, con alta probabilità (nel parametro n) ogni scatola contiene almeno $m/(2n)$ palline. (*Suggerimento:* si usi dapprima un bound di Chernoff per determinare la probabilità che una scatola **fissata** contenga **meno** di $m/(2n)$ palline...)

Answer: For $1 \leq i \leq n$, let X_i be the random variable denoting the number of balls ending up in the i-th box. Also, for $1 \leq j \leq m$, let Y_i^j be an indicator variable such that $Y_i^j = 1$ if the j-th ball is put into the i-th box. Clearly, the Y_i^j's are independent Bernoulli variables with $\Pr(Y_i^j = 1) = 1/n$. Also, we have that $X_i = \sum_{j=1}^m Y_i^j$, with $\mu = E[X_i] = m/n$. Therefore, we can use Chernoff's bound to upper bound the probability that less than $m/2n = (1 - 1/2)\mu$ balls end up in the i-th box:

$$\Pr\left(X_i < \frac{m}{2n}\right) = \Pr\left(X_i < \left(1 - \frac{1}{2}\right)\mu\right) < e^{-\mu(1/2)^2/2} \leq e^{-(16n \log_e n)/(8n)} = \frac{1}{n^2}$$

In order to prove the desired result, we can now proceed as follows:

$$\begin{aligned}
&\Pr(\text{each box contains at least } m/(2n) \text{ balls}) \\
&= 1 - \Pr(\text{at least one box contains less than } m/(2n) \text{ balls}) \\
&= 1 - \Pr\left(\bigcup_{i=1}^{n} \{X_i < m/(2n)\}\right) \\
&\geq 1 - n/n^2 = 1 - 1/n,
\end{aligned}$$

where the last inequality is obtained by applying the union bound. □

Esercizio 3 [11 punti] Dato un insieme $P = \{p_0, p_1, \ldots, p_{n-1}\}$ di n punti nel piano, si sviluppi e si analizzi un algoritmo che restituisca 1 se esitono in P tre punti collineari e

0 altrimenti. Per ottenere punteggio pieno, la complessità dell'algoritmo risultante deve essere $O(n^2 \log n)$. Algoritmi di complessità superiore, anche se corretti, riceveranno punteggio parziale.
(*Suggerimento:* si usi, senza fornirne il codice, la subroutine SORT_BY_POLAR_ANGLE(p, Q) (vista in classe), che dato un punto p e un insieme di punti Q restituisce la lista dei punti di Q ordinati per angolo polare non decrescente rispetto a un sistema di assi centrato in p.)

Answer: The idea behind the algorithm is very simple. As in Graham's scan, we select the point $p_{min} \in P$ of smallest y-coordinate (breaking ties in favour of points of smaller x-coordinate). Then, we sort $P - \{p_{min}\}$ by nondecreasing polar angle with respect to a system of axes centered in p_{min}, into a list $Q = \langle q_0, q_1, \ldots, q_{n-2} \rangle$. The crucial property enforced by the sorting is that p_{min} belongs to a triple of collinear points if and only if there are two *consecutive* points in the sorted list, say q_i and q_{i+1}, such that segments $\overline{p_{min}q_i}$ and $\overline{p_{min}q_{i+1}}$ are collinear (note that the choice of p_{min} is crucial for this to hold). This property can be easily checked in linear time by scanning the sorted list and checking whether any of the cross products $(q_i - p_{min}) \times (q_{i+1} - p_{min})$ is equal to 0, for $0 \leq i < n - 2$. If the test succeeds, we return 1 and exit. Otherwise, we can be sure that p_{min} does not belong to any triple of collinear points, hence we can eliminate it from P and iterate the above procedure on $P' = P - \{p_{min}\}$, until we either discover a triple of collinear points or are left with at most two points.

The pseudocode of the algorithm follows.

```
COLLINEAR(P)
n ← |P|
if n ≤ 2 then return 0
for m ← n downto 3 do
    * Let P = {p_0 = (x_0, y_0), p_1 = (x_1, y_1), ..., p_{m-1} = (x_{m-1}, y_{m-1})} *
    p_min = (x_min, y_min) ← p_0
    for i ← 1 to m − 1 do
        if (y_i < y_min) or [(y_i = y_min) and (x_i < x_min)]
            then p_min ← p_i
    Q ← SORT_BY_POLAR_ANGLE(p_min, P − {p_min})
    * Let Q = ⟨q_0, q_1, ..., q_{m-2}⟩ *
    for i ← 0 to m − 2 do
        if (q_i − p_min) × (q_{i+1} − p_min) = 0
            then return 1
    P ← P − {p_min}
return 0
```

The correctness of the algorithm follows from the above discussion. Its running time, in the worst case, is dominated by $n - 2$ calls to SORT_BY_POLAR_ANGLE on $O(n)$ points, for a total running time of $O(n^2 \log n)$. □

Algoritmica Avanzata – Compito, 21/9/2005 (Durata: 3h)

Studente (Nome, Cognome): _____

Matricola: _____

Prima Parte: domande di teoria

Si forniscano le risposte alle seguenti due domande teoriche:

1. Si dimostri **formalmente** che P $\subseteq$ NP

   Solution:

   Let $L \in$ P, and let $A_L(x)$ be its polynomial decision algorithm. Consider the following verification algorithm:

   $V_L(x, y)$
   return $A_L(x)$

   Clearly, $V_L(x, y)$ runs in time polynomial in $|x|$, hence also polynomial in $|x| + |y|$. Moreover, $\forall x \in L : V_L(x, \epsilon) = 1$ (hence ϵ is a constant-size certificate for all strings in L) and $\forall x \notin L : \forall y \in \{0,1\}^\star : V_L(x, y) = 0$, hence $L_{V_L} = L$. Therefore, V_L is a polynomial verification algorithm for L, which implies that $L \in$ NP.

2. Si dimostri che ogni $a \in \boldsymbol{Z}_n^\star$ ammette inverso moltiplicativo in $\boldsymbol{Z}_n^\star$.

   Solution:

   We know that $\gcd(a, n) = 1$, therefore, by Bézout equality, there exist $x, y \in \boldsymbol{Z}$ such that $1 = ax + ny$. Observe that the same equality implies that $\gcd(x, n) = 1$, hence $x \in Z_n^\star$. Moreover, $ax \bmod n = (1 - ny) \bmod n = 1$, hence $x = a^{-1}$.

Seconda Parte: risoluzione di problemi

Esercizio 1 [11 punti] Si consideri il seguente problema decisionale:

VERTEX COVER or INDEPENDENT SET (VCoIS):
ISTANZA: $\langle G = (V, E), h, k\rangle$, G grafo non orientato, $1 \leq h, k \leq |V|$.
DOMANDA: G ha un vertex cover di taglia h *oppure* un independent set di taglia k?

Si dimostri che VCoIS è NP-Hard.

Answer: Consider the following straightforward reduction from VERTEX_COVER (V_C):

$$f(\langle G = (V, E), h\rangle) = \langle G' = G, h, |V|\rangle.$$

Clearly, f is computable in polynomial (in fact, linear) time. Let us now show that f is a valid reduction from VC to CoVC. If $\langle G = (V, E), h\rangle \in$ VC, then G has a vertex-cover $\hat{V}$ of size h. Since $G' = G$ the same holds for G', hence $\langle G' = G, h, |V|\rangle = f(\langle G, h\rangle) \in$ CoVC. Viceversa, if $\langle G, h\rangle \notin$ VC, we first observe that E cannot be empty (since a graph made of all isolated vertices admits vertex covers of any size). Clearly, $G' = G$ does not have a vertex cover of size h. Moreover, G' cannot have an independent set of size $|V|$, since such a large independent set implies that $E = \emptyset$. It follows that $\langle G' = G, h, |V|\rangle = f(\langle G, h\rangle) \notin$ CoVC. We have proved that f is a valid polynomial-time reduction from VC to CoVC, hence the latter problem is NP-Hard. □

Esercizio 2 [11 punti] Dato un grafo non diretto $G = (V, E)$ si ricorda che un *matching* $M \subseteq E$ è un insieme di archi che non condividono estremi, ovvero $\forall e_1, e_2 \in M, e_1 \neq e_2 : e_1 \cap e_2 = \emptyset$. Si consideri il problema di determinare un matching di cardinalità massima. Si dimostri che il seguente algoritmo greedy:

```
GREEDY_MATCHING(G = (V, E))
m ← |E|;      M ← ∅
* Sia E = {e_1, e_2, ..., e_m} *
for i ← 1 to m do
    if (∀e ∈ M : e ∩ e_i = ∅)
        then M ← M ∪ {e_i}
return M
```

è un algoritmo di 2-approssimazione per il problema.
(*Suggerimento:* Si ragioni per assurdo partendo dall'ipotesi che il matching restituito dall'algoritmo abbia meno della metà degli archi del matching di cardinalità massima...)

Answer: First observe that clearly GREEDY_MATCHING returns a matching. Moreover, the final set M returned has the additional property that $\forall e \in E - M : M \cup \{e\}$ is not a matching (or otherwise e would have been added to M). Consider now a maximum matching

$M^\star$. Clearly, $|M| \leq |M^\star|$. We now show that $|M| \geq |M^\star|/2$ and the result will follow. For the sake of contradiction, assume that $|M| < |M^\star|/2$. Then, there must be an edge e' in $M^\star$ which does not share endpoints with edges in M. This is true because the edges in M cover at most $2|M| < |M^\star|$ vertices, hence at most $|M^\star| - 1$ edges of $M^\star$ may intersect with one edge in M. But then there is an edge $\bar{e} \in M^\star$ such that $M \cup \{\bar{e}\}$ is a matching, which contradicts the aforementioned property of M. □

Esercizio 3 [10 punti] Si consideri il seguente sistema di equazioni modulari:

$$\begin{cases} x &= 2 \bmod 3 \\ x &= 4 \bmod 5 \end{cases}$$

Utilizzando il teorema cinese dei resti, determinare l'unica soluzione al sistema in $\boldsymbol{Z}_{15}$.
Attenzione: Risposte "secche" senza traccia del procedimento seguito riceveranno 0 punti.

Answer: We use the analytic definition of the bijection between $\boldsymbol{Z}_{15}$ and $\boldsymbol{Z}_3 \times \boldsymbol{Z}_5$ implied by the Chinese Reminder Theorem. We have $m_1 = 15/3 = 5$, $m_2 = 15/5 = 3$, ${m_1}^{-1} \bmod 3 = 2$, ${m_2}^{-1} \bmod 5 = 2$, $c_1 = 5 \cdot 2 = 10$, $c_2 = 3 \cdot 2 = 6$. Hence the (unique) solution to the above system of equations is $a = (2 \cdot 10 + 4 \cdot 6) \bmod 15 = 44 \bmod 15 = 14$, whose correctness can be easily checked directly. □

Algoritmica Avanzata – Compito, 19/6/2006 (Durata: 3h)

Studente (Nome, Cognome): _____

Matricola: _____

Prima Parte: domande di teoria

Si dimostrino **rigorosamente** i seguenti due enunciati:

1. $\langle G = (V, E), k\rangle \in$ VERTEX COVER $\Leftrightarrow \langle G^c = (V, E^c), |V| - k\rangle \in$ CLIQUE.

   Solution:

   $\langle G = (V, E), k\rangle \in$ VERTEX COVER
   $\Leftrightarrow \exists V' \subseteq V, |V'| = k : \forall u, v \in V, u \neq v : \{u, v\} \in E \Rightarrow (u \in V') \vee (v \in V')$
   (perform the change of variable $V'' = V - V'$)
   $\Leftrightarrow \exists V'' \subseteq V, |V''| = |V| - k : \forall u, v \in V, u \neq v : \{u, v\} \in E \Rightarrow (u \notin V'') \vee (v \notin V'')$
   (use the counter-positive proposition)
   $\Leftrightarrow \exists V'' \subseteq V, |V''| = |V| - k : \forall u, v \in V, u \neq v : \neg[(u \notin V'') \vee (v \notin V'')] \Rightarrow \{u, v\} \notin E$
   $\Leftrightarrow \exists V'' \subseteq V, |V''| = |V| - k : \forall u, v \in V, u \neq v : (u \in V'') \wedge (v \in V'') \Rightarrow \{u, v\} \in E^c$
   $\langle G^c = (V, E^c), |V| - k\rangle \in$ CLIQUE

2. Dati $a > b \geq 1$, se EUCLID(a, b) esegue $k \geq 1$ chiamate a se stesso (compresa quella esterna) allora $a \geq F_{k+2}$ e $b \geq F_{k+1}$, dove F_i è l'i-simo numero di Fibonacci, con $i \geq 1$.

   Solution:

   The proof is by induction on $k \geq 1$. For the basis $k = 1$, we have $b \geq 1 = F_2$ and $a > b$, hence $a \geq 2 = F_3$. Let the thesis hold for $k - 1$ calls. For k calls, EUCLID(a, b) calls EUCLID$(b, a \bmod b)$, and the latter performs $k - 1$ calls (including the outer one). Therefore, by the inductive hypothesis, we get $b \geq F_{(k-1)+2} = F_{k+1}$ and $a \bmod b \geq F_{(k-1)+1} = F_k$. Moreover, we have $a \bmod b = a - \lfloor a/b \rfloor b$, hence $a = \lfloor a/b \rfloor b + a \bmod b \geq b + a \bmod b \geq F_{k+1} + F_k = F_{k+2}$, and the thesis follows.

Seconda Parte: risoluzione di problemi

Esercizio 1 [11 punti] Dato un grafo non orientato $G(V, E)$ con $|V| > 0$ pari e un sottoinsieme $V' \subset V$ con $|V'| = |V|/2$, l'insieme di archi $B_G(V') = \{\{u, v\} \in E : (u \in V') \wedge (v \in V - V')\}$ è un particolare taglio, detto *bisezione di G rispetto a V'*. Si considerino i seguenti problemi decisionali:

MIN-BISECTION:
I: $\langle G = (V, E), k \rangle$,
$|V| > 0$ pari, $0 \leq k \leq |V|^2/4$
D: $\exists V' \subset V, |V'| = |V|/2 :$
$|B_G(V')| \leq k$?

MAX-BISECTION:
I: $\langle G = (V, E), k \rangle$,
$|V| > 0$ pari, $0 \leq k \leq |V|^2/4$
D: $\exists V' \subset V, |V'| = |V|/2 :$
$|B_G(V')| \geq k$?

Si noti che i due problemi sono i problemi decisionali relativi, rispettivamente, alla minimizzazione e alla massimizzazione della bisezione di un grafo.

Si dimostri che MIN-BISECTION $<_P$ MAX-BISECTION

Answer:

Let $G = (V, E)$ be an undirected graph with $|V| > 0$ even, and let $V' \subset V$ with $|V'| = |V|/2$. It is immediate to argue that $B_{G^c}(V') = |V|^2/4 - B_G(V')$ since the set of all possible (undirected) edges between V' and $V - V'$ contains exactly $|V|^2/4$ elements, and each such edge is either in $B_G(V')$ or in $B_{G^c}(V')$. Therefore G has a bisection of size at most k if and only if G^c has a bisection of size at least $|V|^2/4 - k$. This proves that the function

$$f(\langle G = (V, E), k \rangle) = \langle G^c = (V, E^c), |V|^2/4 - k \rangle$$

which is clearly computable in polinomial (at most quadratic) time, reduces MIN-BISECTION to MAX-BISECTION. Observe that the same function also reduces MAX-BISECTION to MIN-BISECTION. □

Esercizio 2 [10 punti] Si determini il rappresentante principale della classe inversa moltiplicativa di $[14]_{243}$ in $\boldsymbol{Z^\star_{243}}$. **Attenzione:** Si mostri l'intero procedimento.

Answer: It suffices to run the Extended Euclid algorithm (EE) on 243 and 14 to obtain the coefficients x and y of the integer combination $\gcd(243, 14) = 1 = x \cdot 243 + y \cdot 14$. The principal representative of the multiplicative inverse of $[14]_{243}$ will then be y mod 243. The recursive calls generated by EE(243,14) are:

$$\text{EE}(243, 14) \rightarrow \text{EE}(14, 5) \rightarrow \text{EE}(5, 4) \rightarrow \text{EE}(4, 1) \rightarrow \text{EE}(1, 0).$$

The return values (in bottom-up order) are:

$$\{1, (1, 0)\} \rightarrow \{1, (0, 1)\} \rightarrow \{1, (1, -1)\} \rightarrow \{1, (-1, 3)\} \rightarrow \{1, (3, -52)\},$$

hence the sought answer is $-52 \bmod 243 = 243 - 52 = 191$ □

Esercizio 3 [11 punti] Sia S un insieme di n interi positivi distinti, e sia WORK(S) una procedura deterministica che, dato in ingresso S, ritorna un valore intero eseguendo $T_W(|S|) = T_W(n) = n^2$ operazioni. Si consideri ora il seguente algoritmo randomizzato:

```
RAND_REC(S)
if |S| ≤ 1 then return 1
x ← WORK(S); p ← RANDOM(S)
⋆ S_1 = {s ∈ S : s < p}; S_2 = {s ∈ S : s > p} ⋆
if (|S_1| ≥ |S_2|) then y ← RAND_REC(S_1)
                   else  y ← RAND_REC(S_2)
return x + y
```

Si dimostri che $\forall S : |S| = n$, il numero di operazioni eseguito da RAND_REC(S) è $O(n^2 \log n)$ con alta probabilità.

Answer:

Let S be an arbitrary set of n integers. The recursion tree associated to RAND_REC(S) is a unary tree, that is, a simple path. The instance S^i associated to the i-th node of the path has size at most n, hence contributes work $O(|S^i|^2) = O(n^2)$. Therefore, it suffices to show that the path has length $O(\log n)$, with high probability.

Let $m_i = |S^i|$, and let $x_1, x_2, \ldots, x_{m_i}$ be the elements of S^i taken in increasing order. As argued in the analysis of QUICKSORT, if RANDOM(S^i) selects any value x_j with $\lfloor m_i/4 \rfloor + 1 \leq j \leq \lceil 3/4 m_i \rceil$, which occurs with probability (at least) $1/2$, then the size of S^{i+1} is reduced by a factor at least $3/4$ over the size of S^i. Therefore $\lceil \log_{4/3} n \rceil$ successes are sufficient to reach the leaf of the path. (In what follows, for simplicity, floors and ceilings are omitted.)

Consider now event F= "the depth of the recursion tree is greater than $t = a \log_{4/3} n$" where $a > 1$ is a constant to be determined by the analysis. If F occurs, then less than $\log_{4/3} n$ successes occurred among the first t nodes of the tree. Let X_i be the indicator variable signalling that a success has occurred at the i-th node. Clearly, $\Pr(X_i = 1) = 1/2$, and the X_i's are independent. Let now $X = \sum_{i=1}^{t} X_i$ and observe that $\mu = E[X] = t/2 = (a/2) \log_{4/3} n$. Then,

$$\Pr(F) \leq \Pr(X < \log_{4/3} n) = \Pr(X < (1-\delta)\mu) < e^{-\mu\delta^2/2},$$

where we must make sure that $0 < \delta = 1 - 2/a < 1$. Choosing $a = 8$ we obtain $\delta = 3/4$ hence

$$e^{-\mu\delta^2/2} < e^{-\ln n^{1/\ln(4/3)}} = 1/n^{1/\ln(4/3)} = o(1/n^3).$$

Therefore $T(n) = O(n^2 \log n)$, with high probability. □

Algoritmica Avanzata – Compito, 3/7/2006 (Durata: 3h)

Studente (Nome, Cognome): _____

Matricola: _____

Prima Parte: domande di teoria

Si dimostrino **rigorosamente** i seguenti due enunciati:

1. $\forall L \in \{0,1\}^\star : (L <_P L^c) \Leftrightarrow (L^c <_P L)$

   Solution:

   $$\begin{aligned} L <_P L^c &\Leftrightarrow \exists f(x) \text{ p.t.c. : } x \in L \Leftrightarrow f(x) \in L^c \\ &\Leftrightarrow \exists f(x) \text{ p.t.c. : } x \notin L \Leftrightarrow f(x) \notin L^c \\ &\Leftrightarrow \exists f(x) \text{ p.t.c. : } x \in L^c \Leftrightarrow f(x) \in L \\ &\Leftrightarrow L^c <_P L \end{aligned}$$

2. Se il multigrafo $\mathcal{G} = (V, \mathcal{E})$ ha un taglio minimo di cardinalità t, allora $|\mathcal{E}| \geq t|V|/2$.

   Solution:

   For each node $v \in V$, the multiset $\mathcal{E}_v = \{\!\{ \{u, v\} \in \mathcal{E} \}\!\}$ of edges incident on v is a cut of size $|\mathcal{E}_v|$ disconnecting $\{v\}$ from $V - \{v\}$: since t is the size of the minimum cut, it must be $|\mathcal{E}_v| \geq t$. Moreover, observe that $\mathcal{E} = \bigcup_{v \in V} \mathcal{E}_v$, with each edge $\{u, v\}$ occurring twice in the union (once in $\mathcal{E}_u$ and once in $\mathcal{E}_v$). Therefore:

   $$|\mathcal{E}| = \sum_{v \in V} |\mathcal{E}_v|/2 \geq \sum_{v \in V} t/2 = t|V|/2.$$

Seconda Parte: risoluzione di problemi

Esercizio 1 [11 punti] Si consideri il seguente problema decisionale:

NOT-ALL-EQUAL 4-CNF-SAT (NAE-4-CNF-SAT):

I: $\langle \Phi(x_1, x_2, \ldots, x_n) = C_1 \wedge C_2 \wedge \cdots \wedge C_m \rangle$, Φ in forma 4-CNF.

D: Esiste un assegnamento di valori di verità $\boldsymbol{b} \in \{0,1\}^n$ sotto il quale ogni clausola di Φ contiene almeno un letterale vero e almeno un letterale falso?

Si dimostri che NAE-4-CNF-SAT $\in$ NPH.

(*Suggerimento:* si riduca da 3-CNF-SAT. Data $\langle \Phi(x_1, \ldots, x_n) \rangle$ la riduzione fa uso di una nuova variabile $x_{n+1} \ldots .$) .)

Answer: Our reduction f is from 3-CNF-SAT and is defined as follows. Let

$$\Phi(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_m$$

be the 3-CNF-SAT instance. Then

$$f(\Phi) =< \Phi'(x_1, \ldots, x_n, x_{n+1}) = C'_1 \wedge C'_2 \wedge \cdots \wedge C'_m \rangle, \text{ with } C'_i = C_i \vee x_{n+1}, 1 \leq i \leq m.$$

Function f is clearly computable in polynomial (indeed, linear) time. Let us now show that f is a valid reduction. Assume that $\Phi(x_1, \ldots, x_n) \in$ 3-CNF-SAT. Then, there exists a truth assignment $\boldsymbol{b} \in \{0,1\}^n$ which satisfies Φ. Since Φ is a CNF formula, this implies that under $\boldsymbol{b}$ each clause contains a true literal. But then, under $\boldsymbol{b'} = (\boldsymbol{b}|0)$, each clause in Φ' has one true literal and one false one (x_{n+1}), hence $f(\langle \Phi \rangle) = \Phi' \in$ NAE-4-CNF-SAT. Vice versa, let $\Phi' \in$ NAE-4-CNF-SAT, and let $\boldsymbol{b'}$ be the truth assignment under which each clause in Φ' has one true literal and one false one. If $b'_{n+1} = 0$, then it must be that $\Phi(b'_1, \ldots, b'_n) = 1$, since each clause must also contain a true literal under $\boldsymbol{b'}$, hence $\langle \Phi \rangle \in$ 3-CNF-SAT. If $b'_{n+1} = 1$, then consider $\boldsymbol{b''} = (\neg b'_1, \ldots, \neg b'_n, 0)$, and observe that under $\boldsymbol{b''}$ each clause in Φ' *still* has one false literal and one true literal (which must necessarily be one of the original literals of Φ). Then again $\Phi(b''_1, \ldots, b''_n) = 1$, whence $\langle \Phi \rangle \in$ 3-CNF-SAT. □

.

Esercizio 2 [10 punti] Sia $n = p \cdot q$, con n noto e p, q incogniti. Si supponga che un criptoanalista a conoscenza di n sia in grado di invocare una routine RESIDUE(x) che, dato in ingresso un arbitrario valore $x \in \boldsymbol{Z}_n$ ritorna il valore x mod p. Si scriva un algoritmo FACTOR(n) che, invocando RESIDUE **una sola volta** su un dato valore, sia in grado di ritornare la coppia (p, q).

Answer: It is sufficient to observe that $(n-1) \bmod p = (n \bmod p + (-1) \bmod p) \bmod p = p-1$, therefore the single call RESIDUE($n-1$) returns $x = p-1$, hence we may compute $p = x+1$ and $q = n/(x+1)$. The algorithm follows.

FACTOR(n)
$x \leftarrow$ RESIDUE($n-1$)
return $(x+1, n/(x+1))$

□

Esercizio 3 [11 punti] Si consideri la versione di ottimizzazione del problema NAE-4-CNF-SAT definito nell'Esercizio 1: data una formula 4-CNF, si vuole determinare il **massimo** numero di clausole che contengono sia un letterale vero che un letterale falso sotto un qualche assegnamento di valori verità alle variabili della formula. Si fornisca lo pseudocodice e si analizzi un algoritmo randomizzato polinomiale di $\boldsymbol{\rho}$-approssimazione per il problema e si valuti $E[\boldsymbol{\rho}]$.

Answer: We use the same strategy seen in class for MAX-3SAT: namely, given a 4-CNF formula $\Phi(x_1, x_2, \ldots, x_n) = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, we pick a random truth assignment $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ and return the number of clauses which contain both a true and a false literal under $\boldsymbol{b}$. The code follows.

RAND-APPROX-MAX-NAE-4-CNF-SAT($\langle \Phi(x_1, x_2, \ldots, x_n) \rangle$)
$\star$ Let $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$: $C_i = y_1^i \vee y_2^i \vee y_3^i \vee y_4^i$, $1 \leq i \leq m$. $\star$
for $j \leftarrow 1$ **to** n **do** $b_j \leftarrow$ RANDOM($\{0,1\}$)
$X \leftarrow 0$
for $i \leftarrow 1$ **to** m **do**
  if $(\exists h, k \in \{1,2,3,4\} : (y_h^i|_{\boldsymbol{b}} = 0) \wedge (y_k^i|_{\boldsymbol{b}} = 1))$
    then $X \leftarrow X + 1$
return X

The above algorithm is clearly polynomial (indeed, linear) in the size of the formula. Let us analyze its approximation ratio. First, observe that when $\Phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, the optimum value $k^\star$ cannot be larger than m. Let $\boldsymbol{\rho}$ be the random variable denoting the approximation ratio achieved by the algorithm. We have that

$$E[\boldsymbol{\rho}] = E\left[\frac{k^\star}{\boldsymbol{X}}\right] \leq \frac{m}{E[\boldsymbol{X}]},$$

where $\boldsymbol{X}$ is the random variable denoting the value returned by the algorithm[1].

Let X_i be an indicator variable whose value is one iff C_i contains both a true and a false literal under the random assignment $\boldsymbol{b}$. Clearly,

$$\boldsymbol{X} = \sum_{i=1}^{m} X_i \text{ and } E[\boldsymbol{X}] = \sum_{i=1}^{m} E[X_i] = \sum_{i=1}^{m} \Pr(X_i = 1).$$

Let us now evaluate $\Pr(X_i = 1)$, for $1 \leq i \leq m$. Recall that the literals $y_1^i, y_2^i, y_3^i, y_4^i$ in clause C_i come from four distinct variables $x_1^i, x_2^i, x_3^i, x_4^i$, hence under a random assignment

[1] In fact, one should observe that the random variable X may return value 0, which would make the above expression meaningless. However, we can modify the algorithm so that it returns $\max\{1, X\}$, since there is always a way to pick an assignment under which at least a clause has one false and one true literal. This modification takes care of the problem and only improves the approximation ratio of the resulting algorithm.

they assume value 0 or 1 independently of each other. Out of the 16 different assignments to the 4 variables, there are only two assignments which yield $y_1^i = y_2^i = y_3^i = y_4^i$, hence $\Pr(X_i = 1) = 1 - 2/16 = 7/8$. As a consequence

$$E[\boldsymbol{X}] = \frac{7m}{8}, \text{ hence } E[\boldsymbol{\rho}] \leq \frac{m}{(7/8)m} = \frac{8}{7}.$$

□