Algorithm HW#2

B02901178 江誠敏

May 15, 2017

1 Problem 1.

Consider the boolean expression

$$(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$$

The corresponding linear programming is:

$$\begin{array}{llll}
\text{maximize} & z_1 + z_2 + z_3 + z_4 & & \\
\text{s.t.} & z_1 & \leq & y_1 \\
& z_1 & \leq & y_2 \\
& z_2 & \leq & 1 - y_1 \\
& z_2 & \leq & y_2 \\
& z_3 & \leq & y_1 \\
& z_3 & \leq & 1 - y_2 \\
& z_4 & \leq & 1 - y_1 \\
& z_4 & \leq & 1 - y_2 \\
& z_j & \leq & 1 \qquad \forall j \\
& y_i, z_j & \geq & 0 \qquad \forall i, j
\end{array}$$

It is easy to check that $(y_1, y_2, z_1, z_2, z_3, z_4) = (0.5, 0.5, 1, 1, 1, 1)$ is a solution with $z_1 + z_2 + z_3 + z_4 = 4$. But obviously at most 3 clauses could be satisfied, since exactly one of $x_1, \neg x_1$ would be false, say x_1. Similarly, assume x_2 is false, then $x_1 \vee x_2$ is false. Thus the integrality gap is $3/4$.

2 Problem 2.

First we prove two simple lemmas:

Lemma 1. $1 - 4^{y-1} \leq 4^{-y}$.

Proof. Since

$$4^{-y} - 1 + 4^{y-1} = (2^{-y} - 2^{y-1})^2 \geq 0 \implies 1 - 4^{y-1} \leq 4^y$$

□

Lemma 2. $1 - 4^{-z} \geq 3z/4$.

Proof. $f = z \mapsto 1 - 4^{-z}$ is concave since $f'' = -4^{-z}(\log 4)^2 < 0$.

$$4^{-y} - 1 + 4^{y-1} = (2^{-y} - 2^{y-1})^2 \geq 0 \implies 1 - 4^{y-1} \leq 4^{y}$$

Thus

$$1 - 4^{-z} f(z) = f((1-z)\cdot 0 + z \cdot 1) \geq (1-z)f(0) + zf(1) = \frac{3z}{4}$$

□

Now, for each clause, let z_i be the corresponding variable in the LP, $A \triangleq \{\, i : x_i \text{ appears in the clause}\,\}$ and $B \triangleq \{i : \neg x_i \text{ appears in the clause}\}$ (Notice that A, B might intersect). Then we have

$$z_i \leq \sum_{i\in A} y_i + \sum_{i \in B} 1 - y_i$$

For each $i \in A$, x_i would be assigned "`True`" with probability $4^{y_i - 1}$, and thus "failed" with probability $1 - 4^{y_i-1}$. Similarly, for each $i \in B$, x_i would be assigned "`False`" with probability $1 - 4^{y_i-1}$, and thus "failed" (i.e., $\neg x_i =$ `False`) with probability 4^{y_i-1}. Thus the overall probability of "success" (i.e., the clause evaluate to "True") is

$$\begin{aligned}
1 - \prod_{i\in A}(1 - 4^{y_i-1}) \prod_{i\in B} 4^{y_i-1} &\overset{\text{(a)}}{\geq} 1 - \prod_{i\in A} 4^{-y_i} \prod_{i \in B} 4^{y_i-1} \\
&\geq 1 - 4^{\left(\sum_{i\in A} -y_i + \sum_{i\in B} -(1-y_i)\right)} \\
&\geq 1 - 4^{-z_i} \\
&\overset{\text{(b)}}{\geq} \frac{3z_i}{4}
\end{aligned}$$

Where Lemma 1 and 2 is directly used in (a) and (b). Thus the expected weights of clause that is satisfied is greater then $3\sum w_i z_i/4$, hence the rounding provides an 4/3-approximation algorithm.

3 Problem 3.

If (S_1, S_2) are a partition of vertices such that $S_1 \cup S_2 = V$ and $S_1 \cap S_2 = \varnothing$, Define the cut (edges) as $C = \{\, (u, v) : u \in S_1 \text{ and } v \in S_2\,\}$.

Consider the following randomized algorithm: Simply place each u randomly into $\tilde{S}_1$ or $\tilde{S}_2$ with equal probability and independently, then output $(\tilde{S}_1, \tilde{S}_2)$. If C_{opt} is the optimal cut, for each $(u, v) \in C_{\text{opt}}$, $\Pr\left\{\, u \in \tilde{S}_1 \text{ and } v \in \tilde{S}_2 \,\right\} = 1/4$, thus $\mathbf{E}|\tilde{C}| \geq 1/4|C_{\text{opt}}|$, where $\tilde{C}$ is the cut induced by $(\tilde{S}_1, \tilde{S}_2)$.

Lemma 3. *If some vertices is already assigned, then the expected size of the cut if the remaining vertices are assigned to S_1 or S_2 randomly could be calculated in $\mathcal{O}(|E|)$.*

Proof. We simply examinate each cases carefully, for each edge $e = (u, v)$, the probability p of e be in the cut is:

1. If both u, v are already assigned: If $u \in S_1$ and $v \in S_2$, then $p = 1$. Otherwise, $p = 0$.
2. If u is already assigned: If $u \in S_1$, then $p = 0.5$. Otherwise, $p = 0$.
3. If v is already assigned: If $v \in S_2$, then $p = 0.5$. Otherwise, $p = 0$.
4. Else, $p = 0.25$.

Now, by the additivity of expectation, $\mathbf{E}|C| = \sum_{e \in E} \Pr\{ e \in C \}$. Thus we could compute $\mathbf{E}|C|$ by running through all edges, which obviously cost $\mathcal{O}(|E|)$. $\square$

Now, we derandomize the algorithm by following: For i from 1 to $|V|$,

1. Calculate the expected value $E_{i,1}, E_{i,2}$ by lemma 3, where $E_{i,j}$ is the expected cut size when v_i is assigned to S_j.
2. Let $E_{i,j}$ be the greater one, then assign v_i to S_j.

Let x_i be the expected cut size after v_i is assigned, then in the very beginning, $x_0 = \mathbf{E}|\tilde{C}| \geq |C_{\text{opt}}|/4$. Now, if $x_{i-1} \geq |\tilde{C}|$, then when assigning v_i, we must have $x_{i-1} = 0.5E_{i,1} + 0.5E_{i,2}$, thus one of them must be greater then x_{i-1}, thus $x_i \geq x_{i-1} \geq \mathbf{E}|\tilde{C}| \geq |C_{\text{opt}}|/4$ by induction, hence derandomizing gives a deterministic 4-approximation algorithm. Since we iterate through all the vertices, and each time applying Lemma 3 cost $\mathcal{O}(|E|)$, the overall running time is $\mathcal{O}(|E||V|)$. Notice that the "type" of an edge (listed 1.— 4. above) changed at most 2 time (once u, v are assigned). It is surely possible to reduce the running time to $\mathcal{O}(|E| + |V|)$ by calculating only the edge which type changed in Lemma 3.

4 Problem 4.

Let m, s be the current number of elements and the size of the hash table, respectivily (so $s = 4k^2$, where we rebuilt the hash table while inserting the k-th element from the statement). Define the potential function as $\Phi \triangleq 2m - \sqrt{s} = 2m - 2k$. Since k is the last time we rebuilt the hash table, $m \geq k = \sqrt{s}/2$ and hence $\Phi = 2m - \sqrt{s} \geq 0$ at any moment.

Let m', s' be the value of m, s after an insertion, there are two cases.

1. No collision happened, then $\Delta\Phi = 2\Delta m = 2(m' - m) = \mathcal{O}(1)$, thus the amortized cost is $\tilde{c} = c + \Delta\Phi = \mathcal{O}(1) + \mathcal{O}(1) = \mathcal{O}(1)$.
2. A collision occurred. Since the hash function is chosen from a 2-universal hash family, this happened with probability no more then m/s. After rebuilding, $s' =$

$4k'^2 = 4m^2$. Thus we have

$$\Delta\Phi = 2\Delta m - \Delta s = \mathcal{O}(1) + \sqrt{s'} - \sqrt{s} = \mathcal{O}(1) + 2k - 2m$$

Hence the amortized cost is

$$\tilde{c} = c + \Delta\Phi = m + \mathcal{O}(1) + 2k - 2m = 2k - m + \mathcal{O}(1)$$

Now, let C be the amortized cost of an insertion, by the discussion above, we have

$$\mathbf{E}\, C = \mathcal{O}(1) + \frac{m}{s}(2k - m) = \mathcal{O}(1) + \frac{m(2k-m)}{4k^2}$$

If $m > 2k$, then $2k - m$ is negative, so $\mathbf{E}\, C = \mathcal{O}(1)$. Thus assume $m \le 2k$, then

$$\frac{m(2k-m)}{4k^2} \le \frac{2km}{4k^2} \le \frac{4k^2}{4k^2} = \mathcal{O}(1) \implies \mathbf{E}\, C = \mathcal{O}(1)$$

Thus no matter what, $\mathbf{E}\, C = \mathcal{O}(1)$. Hence the expected cost of inserting n elements is $n\,\mathcal{O}(1) = \mathcal{O}(n)$.

5 Problem 5

Assuming n is the number of a_is.

Divide the interval $[0, 1)$ into $[0, m), [m, 2m), \ldots, [km, (k+1)m), \ldots, [(1/m-1)m, 1)$ where $m = 1/n^2$. Let $I_k \triangleq [km, (k+1)m)$. Now,

$$\begin{aligned}
&\Pr\{\text{ the smallest interval} \le 1/n^2 = m\,\} \\
&= \Pr\{\text{ exists one interval} \le m\,\} \\
&\ge \Pr\{\,\exists i \ne j \text{ s.t. } a_i \text{ and } a_j \text{ are both in } I_k \text{ for some } k\,\} \\
&= 1 - \Pr\{\,\forall\, k, I_k \text{ contains no more then one } a_i\,\} \triangleq 1 - p
\end{aligned}$$

So we shall calculate p, which is the probability of every I_k is occupied by at most one a_i. Since these a_is are picked uniformly and independently at random, and each of them has size m, $\Pr\{\, a_i \in I_k \,\} = m = 1/n^2$. Notice that

$$\begin{aligned}
p &= \prod_{i=1}^{n} \Pr\{\, a_i \text{ is inserted into an empty sub-interval}\,\} \\
&= \prod_{i=1}^{n} \left(1 - \frac{i-1}{n^2}\right) = \prod_{i=0}^{n-1} \left(1 - \frac{i}{n^2}\right)
\end{aligned}$$

Since before a_i is inserted, $i - 1$ sub-intervals are already occupied by the previous a_js, provided that no collision occurred before.

Lemma 4. *$e^{-x} \ge 1 - x$, $\forall x \in \mathbb{R}$.*

Proof. Let $f = e^{-x} - (1-x)$, then $f(0) = 0$ and $f'(x) = 1 - e^{-x} \lesseqgtr 0$ as $x \lesseqgtr 0$. We conclude that $e^{-x} \geq 1 - x$. $\square$

By lemma 4,

$$p \leq \prod_{i=0}^{n-1} \exp\left(-\frac{i}{n^2}\right) = \exp\left(-\sum_{i=0}^{n-1} \frac{i}{n^2}\right) = \exp\left(-\frac{n(n-1)}{2n}\right) = \exp\left(-\frac{1}{2} + \frac{1}{2n}\right)$$

If $n \geq 2$, $-1/2 + 1/(2n) \leq 1/4$, thus

$$p \leq \exp\left(-\frac{1}{2} + \frac{1}{2n}\right) \leq e^{-1/4} = \Omega(1), \quad \text{as } n \geq 2.$$

Thus the size of the smallest interval is less then $1/n^2$ has probability no less then $e^{-1/4} = \Omega(1)$.