# Homework 2


## 吴昊 2016K8009929015（本科生）

## 邮件：wuhao164@mails.ucas.ac.cn


# 1 successful prob of Fast-Cut

Successful probablity:
the origin size n problem is reduced to two subproblem size $\frac{n}{\sqrt{2}}$, considering the contract procedure, the probability that the result of subproblem is exactly the result of origin problem is

$$(1-\frac{2}{n})(1-\frac{2}{n-1})...(1-\frac{2}{n/\sqrt{2}})=\frac{(\frac{n}{\sqrt{2}}-1)(\frac{n}{\sqrt{2}}-2)}{n(n-1)}\geq\frac{1}{2}$$

denote the successful probabilty of origin problem with size n by $P(k+1)$, subproblem with size $\frac{n}{\sqrt{2}}$ by $P(k)$, then we have

$$P(k+1)=1-(1-\frac{1}{2}P(k))^2$$

so we get

$$P(k+1)=P(k)-\frac{1}{4}P(k)^2$$

perform a change of variable $P(k)=\frac{4}{q(k)+1}$, yields the following simplification:

$$q(k+1)=q(k)+1+\frac{1}{q(k)}$$

recursively apply the equation to the righthand side

$$q(k)=q(1)+k-1+\sum_{1\leq i\leq k-1}\frac{1}{q(k)}\geq k-1$$

by using $q(k)\geq k-1$,we have

$$q(k) \le q(1) + k - 1 + \sum_{1 \le i \le k-1} \frac{1}{i} \le q(1) + k - 1 + \log k$$

combine the upper bound and lower bound

$$q(k) = \Omega(k)$$

$$P(k) = \Omega(\frac{1}{k})$$

observe that $k = \log n$,we can draw the conclution

$$Pr(n) = \Omega(\frac{1}{\log n})$$

# 2 Exercise 10.9

the time complexity of the modified algorithm is $\Theta(n^2 + t^3)$ with reducing procedure costs $\Theta(n^2)$ and cubic time algorithm for subproblem.

consider the successful probability,after reducing the problem,the probability that the subproblem result is exactly the origin problem result is

$$(1 - \frac{2}{n})(1 - \frac{2}{n-1})...(1 - \frac{2}{n/\sqrt{2}}) = \frac{t(t-1)}{n(n-1)} = \Theta(\frac{t^2}{n^2})$$

to ensure at least 1/2 successful probabilty,repeat the algorithm $\Theta(\frac{n^2}{t^2})$times

the total running time

$$\Theta(n^2 + t^3) * \Theta(\frac{n^2}{t^2}) = \Theta(\frac{n^4}{t^2} + n^2 t) = \Omega(n^{\frac{8}{3}})$$

the last step above is by using inequality of geometric and arithmetic means

$$\frac{n^4}{t^2} + \frac{n^2 t}{2} + \frac{n^2 t}{2} \ge (\frac{n^8}{4})^{\frac{1}{3}}$$

# Optional(k-way cut-set)

Denote the minimum k-way cut-set is S,the size of the k-way cut-set is |S|. Sum over all possible trival k-way (k-1 singletons and the complement), and count how many times an edge is over-counted,we have

$$\binom{n}{k-1}|S| \le [\binom{n-2}{k-3} + 2\binom{n-2}{k-2}]|E|$$

Notice that when an edge is counted in the cut-set, its two endpoints are either *both singletons* or *one singletons and one in the complement*.
Simplify the inequality and get

$$|E| \ge \frac{n(n-1)}{(2n-k)(k-1)}|S| \ge \frac{n}{2(k-1)}|S|$$

Run the Min-Cut algorithm on the graph until $2k-1$ vertices left, the probability that cut-set of the left $2k-1$ vertices is the answer of the original problem is

$$\begin{aligned}(1-\frac{2(k-1)}{n})(1-\frac{2(k-1)}{n-1})...(1-\frac{2(k-1)}{2k+1})(1-\frac{2(k-1)}{2k})\\ =\frac{2k(2k-1)...3\cdot 2}{n(n-1)...(n-2k+4)(n-2k+3)}\\ \ge(\frac{1}{n})^{2k-4}\end{aligned}$$

Actually,the successful probabilty could be much larger,but $(\frac{1}{n})^{2k-4}$ is enough for analysis
Since the run time is $\Theta(n^2)$,to abtain a constant successful probabilty,we can repeat the algorithm $O(n^{2k-4})$ times. So the time complexity comes to $O(n^{2k-2})$ .

# Exercise 1.10 Page 22

Assume that the algorithm is A,and let $\epsilon = \frac{1}{2^n} \forall x \in \Sigma^*, \begin{cases} x \in L \Rightarrow Pr(A(x)accepts) = \frac{1}{2}+\epsilon \\ x \notin L \Rightarrow Pr(A(x)accepts) = \frac{1}{2}-\epsilon \end{cases}$

Repeat the algorithm $p$ times (p is a polynomial of n) and adopt the major vote strategy, if the error probability is $\frac{1}{4}$,

$$
\begin{aligned}
&\sum_{1\leq i\leq \frac{p}{2}} \binom{p}{i}[(\frac{1}{2}+\epsilon)^i(\frac{1}{2}-\epsilon)^{p-i}-(\frac{1}{2}-\epsilon)^i(\frac{1}{2}+\epsilon)^{p-i}] \\
&\leq \sum_{1\leq i\leq \frac{p}{2}} \binom{p}{i}(\frac{1}{2})^{p-i}[(\frac{1}{2}+\epsilon)^i-(\frac{1}{2}-\epsilon)^i] \\
&\leq \sum_{1\leq i\leq \frac{p}{2}} \binom{p}{i}(\frac{1}{2})^{p-i}4i(\frac{1}{2})^{i-1}\epsilon \\
&\leq \sum_{1\leq i\leq \frac{p}{2}} \binom{p}{i}(\frac{1}{2})^{n+p-3} \\
&= \frac{poly(n)}{2^n} < \frac{1}{4}
\end{aligned}
$$

# Exercise 1.13 Page 27

**Prove** PP=coPP,
$L \in PP$,$L^c$denote the complement of L. By the definition of PP,there exists an algorithm A,for x in L,Pr(A(x)accepts)>1/2,for x not in L,Pr(A(x)accepts)=0.
construct an algorithm B,B rejects when A accepts,and B accepts when A rejects.
Then

$$
x \in L^c \Rightarrow Pr(B(x)accepts) = 1 - Pr(A(x)accepts) > \frac{1}{2}
$$
$$
x \notin L^c \Rightarrow Pr(B(x)accepts) = 1 - Pr(A(x)accepts) < \frac{1}{2}
$$

which means PP=co-PP

**Prove** BPP=co-BPP
assume $L \in BPP$ with algorithm A,construct algorithm B in the same way above.
Then

$$
x \in L^c \Rightarrow Pr(B(x)accepts) = 1 - Pr(A(x)accepts) > \frac{3}{4}
$$
$$
x \notin L^c \Rightarrow Pr(B(x)accepts) = 1 - Pr(A(x)accepts) < \frac{1}{4}
$$

implys BPP=co-BPP

# Optional Problem 1.15 Page 27

Show that NP $\subseteq$ BPP implies NP = RP