記分 Score	教師簽章 Instructor Signature

國立臺北大學 學年度第 學期期中/末 考試試卷
National Taipei University Student's Answer Sheet

系級/Department & Grade 資工四 科目/ Course Title 110高等演算期中考

(該科目所屬系級)/ Course Given Department 資工碩一

☑學士班 Bachelor Program ☐碩士班 Master Program ☐博士班 Ph.D. Program

學號/ Student ID 姓名/ Student Name

任課教師/ Instructor Name 110 年Year 11 月Month 18 日Date

(上列各項，考生須逐項填明，違者該科試卷以零分計)

Please fill out the above information, otherwise the score of this quiz will be zero.

(第一面) Page1

1. (老師有套固定的寫法，建議參考他的做法)

令 $k=\lfloor \log_3 \log_2 (n-1) \rfloor$

$n=3\sim8$ 時, $a=8$, $b=6$, $c=5$ $\longrightarrow$ Loop 1次, $k=0$

$n=9\sim512$ 時, $a=512$, $b=12$, $c=6$ $\longrightarrow$ Loop 2次, $k=1$ $\rightarrow$ $k+1$ 與 Loop 次數相等

$\vdots$

(a) $a=2^{3^{\lfloor \log_3 \log_2 (n-1) \rfloor+1}}$ (b) $b=3\times 2^{\lfloor \log_3 \log_2 (n-1) \rfloor+1}$ (c) $C=5+\lfloor \log_3 \log_2 (n-1)$ (d) $\Theta(\lg \lg n)$

2.

(a) 令 $c(n)$ 為 input n 時的 count 結果

$\begin{cases} c(n)=c(n-1)+2c(n-2)+2 \\ c(1)=3,\ c(2)=3 \end{cases}$ $\begin{cases} \text{homo}: x^2=x+2, & x=-1 \text{ or } 2 \\ \text{hetro}: x=x+2x+2, & x=-1 \end{cases}$, 令 $c(n)=\alpha 2^n+\beta(-1)^n-1$

將 $C(1)=3$, $C(2)=3$ 代入 $C(n)$, $\alpha=\frac{4}{3}$, $\beta=-\frac{4}{3}$, 故 $C(n)=\frac{4}{3}2^n+\left(-\frac{4}{3}\right)(-1)^n-1$.

(b) 令 $T(n)$ 為 input n 的 time function

$$\begin{cases} T(n)=T(n-1)+2T(n-2)+1 \\ T(1)=1,\ T(2)=1 \end{cases}$$

注意與(a)小題的關係式僅差在 +1 與 +2,

故 $T(n)=\frac{2}{3}2^n-\frac{2}{3}(-1)^n-1$, $T(n)\in\Theta(2^n)$

3,

令 $q_i=(1-p_i)$

(a) $R_L(1,n,k)=p_{n-k+1}q_{n-k+2}\cdots q_nR_L(1,n-k,k)+p_{n-k+2}q_{n-k+3}\cdots q_nR_L(1,n-k+1,k)+\cdots+$
$p_{n-1}q_nR_L(1,n-2,k)+p_nR_L(1,n-1,k)$

(b) $R_L(1,n,k)=R_L(1,n-1,k)-p_{n-k}q_{n-k+1}q_{n-k+2}\cdots q_nR_L(1,n-k-1,k)$

(c) $R_C(1,n,k)=R_L(2,n,k)+R_L(3,n+1,k)+\cdots+R_L(k+1,n+k-1,k)$

4,

(a)

```
int ans = 0, a[n] = {0,1};
for(int i=2; i<=n; i++){
    a[n] = a[n-1] + a[n-2];
}
```

$f(n)=a[n]$

整個過程僅1個for-loop做 n-times iteration,

故 time complexity $=\Theta(n)$

(b)

$$\begin{bmatrix}F(n)\\F(n-1)\end{bmatrix}=\begin{bmatrix}1&1\\1&0\end{bmatrix}\begin{bmatrix}F(n-1)\\F(n-2)\end{bmatrix}=\begin{bmatrix}1&1\\1&0\end{bmatrix}^{n-1}\begin{bmatrix}F(1)\\F(0)\end{bmatrix},$$ 令 $A=\begin{bmatrix}1&1\\1&0\end{bmatrix}$, 則需在 $\Theta(\lg n)$ 內算出 A^n:

$$\begin{cases} A^n=A^{n/2}\cdot A^{n/2}, & \text{if } n=\text{even} \\ A^n=A^{\lfloor n/2\rfloor}\cdot A^{\lfloor n/2\rfloor}\cdot A & \text{otherwise} \end{cases}$$

, 矩陣乘法 $\Theta(1)$, 該 algo 做 $\lg_2(n)$ 次 recursive 會到 A^1,

故整體 time complexity $=\Theta(\lg n)$

5

(a)

```
while(1){
    a=Biased_Rand(); b=Biased_Rand();
    if(a!=b) return a;
}
```

(b)

Case	機率
1, 1	p^2
1, 0	$p(1-p)$
0, 1	$(1-p)p$
0, 0	$(1-p)^2$

$P(\text{Loop } 1 \text{ return } 0) = p(1-p)$, $P(\text{Loop } 1 \text{ return } 1) = p(1-p)$,

$P(\text{Loop again}) = p^2+(1-p)^2$, 令 $q = p^2+(1-p)^2 = 2p^2-2p+1$

$P(\text{return } 0) = p(1-p)(1+q+q^2+\cdots) = \frac{p(1-p)}{1-q} = \frac{p(1-p)}{2p-2p^2} = \frac{1}{2}$

$P(\text{return } 1) = \frac{1}{2}$ ∵過程相同

(c),

令 algo. expected time $= E[X]$, $q=p^2+(1-p)^2$, 跑 1次 Loop 的 time $= C$

$E[X] = c+qc+q^2c+\cdots \frac{c}{1-q} = \frac{c}{2p(1-p)}$

6,

(a)

```
While(1){
    X = 4*Unbiased_Rand() + 2*Unbiased_Rand() + Unbiased_Rand();
    if(X<3) return 0;
    if(3≤X<5) return 1;
}
```

(b)

$X=0 \sim X=7$ 機率均為 $(\frac{1}{2})^3=\frac{1}{8}$，一次loop就return 0的機率：$\frac{3}{8}$，一次loop就return 1的機率：$\frac{2}{8}$，多跑一次的機率：$\frac{3}{8}$，

$P(\text{return } 0) = \frac{3}{8} + \frac{3}{8} \times \frac{3}{8} + (\frac{3}{8})^2 \times \frac{3}{8} \cdots + = \frac{3}{8} \times \frac{1}{1-\frac{3}{8}} = \frac{3}{5}$

$P(\text{return } 1) = \frac{1}{1-\frac{3}{8}} \times \frac{2}{8} = \frac{2}{5}$，algo is correct!

(c)

設跑一次的time為 c，then expected running time $= c + \frac{3}{8}c + \cdots = \frac{8}{5}c = \Theta(1)$

7,

(a) time complexity of counting sort: $O(n+2^r)$

(b) time complexity of radix sort: $O(\frac{b}{r}(n+2^r))$

(c) take $r = \lg_2 n$, $n^5 - 1 = 2^b$, take $b = 5\lg n$

$O(\frac{b}{r}(n+2^r)) = O(\frac{5\lg n}{\lg n}(n+n)) = O(5 \cdot 2n) = O(n)$

8,

(a) 每做一個 comparison 會淘汰一個數字，97個數字要淘汰96個數字才能出現 maximum

96 comparisons are necessary

(b) 承(a)，96 comparisons 會得到一個單淘汰競賽樹，生成 $\lceil 97/2 \rceil = 49$ 個於 first round 就淘汰的 the smaller numbers，再用類似(a)的手法找出 the smallest number，

$96 + (49-1) = 144$ comparisons

(c) 承a，做出競賽樹後，與 the maximum compare 的數字都有可能為 the second largest number，故要

$96 + \lceil \lg_2 97 \rceil - 1 = 102$ comparisons，

9.

令 $T(n)$ 為執行 i-th smallest number 的 time，分成5個一組找 medium of medium，再踢除部分數字後做 recursion，則 time complexity 分析如下：

$$T(n) = \underbrace{O(n)}_{\text{分成5個一組找medium}} + \underbrace{T(n/5)}_{\text{medium of medium}} + \underbrace{O(n)}_{\text{pivot and split}} + \underbrace{T(3n/4)}_{\text{至少踢除}1/4\text{個數字}}$$

設 $T(n) \le cn$，$T(n) \le \frac{1}{5}cn + \frac{3}{4}c(n) + O(n)$

$$= \frac{19}{20}cn + O(n)$$

$$T(n) \le cn - \left(\frac{1}{20}cn - \Theta(n)\right) \le cn$$，QED，

依 substitution method，$T(n) \in \Theta(n)$