- 书面作业讲解
  - CS第5.1节问题6、10、11、12、13
  - CS第5.2节问题2、9、10、14、15
  - CS第5.3节问题3、4、8、11、12、13
  - CS第5.4节问题5、6、8、10、17、20、21

CS第5.1节问题6

- 2 pennies (1 cent), 1 nickel (5 cents), 1 dime (10 cents)
- without replacement

P_1P_2, P_2P_1	p(PP)=1/6
P_1D, P_2D	p(PD)=1/6
DP_1, DP_2	p(DP)=1/6
P_1N, P_2N	p(PN)=1/6
NP_1, NP_2	p(NP)=1/6
ND	p(ND)=1/12
DN	p(DN)=1/12

CS第5.1节问题10

- Probability that a five-card hand is straight
  - By using five-element sets as your model

$$\frac{9\cdot 4^5}{\binom{52}{5}}$$

  - By using five-element permutations as your model

$$\frac{9\cdot(20\cdot 16\cdot 12\cdot 8\cdot 4)}{52^{\underline{5}}}=\frac{9\cdot 4^5\cdot 5!}{\binom{52}{5}\cdot 5!}=\frac{9\cdot 4^5}{\binom{52}{5}}$$

CS第5.2节问题2

- Selected two from eight kings and queens. What is the probability that the king or queen of spades is among the cards selected?
  - $$\frac{\binom{7}{1}+\binom{7}{1}-\binom{2}{2}}{\binom{8}{2}}=\frac{13}{28}$$
  - $$1-\frac{\binom{6}{2}}{\binom{8}{2}}=\frac{13}{28}$$

CS第5.2节问题9

- P270， 公式5.10

CS第5.2节问题10

- P272， Theorem 5.4

CS第5.2节问题14

- P266，Theorem 5.3

$$1-\sum_{k=1}^{n}(-1)^{k+1}\binom{n}{k}\frac{(2n-1-k)!2^k}{(2n-1)!}=\sum_{k=0}^{n}(-1)^{k}\binom{n}{k}\frac{(2n-1-k)!2^k}{(2n-1)!}$$

CS第5.2节问题15

- P272，principle of inclusion and exclusion for counting

$$
\begin{aligned}
&N_a(\varnothing)-\sum_{k=1}^{m}(-1)^{k+1}\sum_{\substack{i_1,i_2,\ldots,i_k\\ 1\le i_1<i_2<\ldots<i_k\le m}}\left|E_{i_1}\cap E_{i2}\cap\ldots\cap E_{i_k}\right|\\
&=N_a(\varnothing)-\sum_{\substack{K\subseteq P\\ K\neq\varnothing}}(-1)^{|K|+1}N_a(K)\\
&=\sum_{K\subseteq P}(-1)^{|K|}N_a(K)
\end{aligned}
$$

- How this formula could be used to compute the number of onto functions.
  - object → function
  - property → location
  - object has a property → function maps nothing to a location
  - #onto_function → $N_e(\emptyset)$
  - $N_a(K)$ → $(m-|K|)^n$

CS第5.3节问题3、4

- 怎么证明两个event相互独立？
  - E∩F=Ø？
  - P(E)=P(F)？

  - 定义：P(E|F)=P(E)
  - 定理5.5：P(E)P(F)=P(E∩F)

CS第5.3节问题12

- The probability that the family has two girls, given that one of the children is a girl
  - (1/4)/(3/4)=1/3

CS第5.3节问题13

- Monty Hall problem
  - http://en.wikipedia.org/wiki/Monty_Hall_problem

  - P(switch and win)=2/3
  - P(not switch and win)=1/3

CS第5.4节问题6

- Expected sum of the tops of n dice
  - $E(X_1+\ldots+X_n)=E(X_1)+\ldots+E(X_n)=3.5n$

CS第5.4节问题8

- choose 26 cards from 52
- Is the event of having a king on the ith draw independent of the event of having a king on the jth draw?
  - P(i)=P(j)=1/13
  - $\frac{A_4^2}{A_{52}^2} = \frac{1}{13 \cdot 17} \neq P(i) \cdot P(j)$
- How many kings do you expect to see?
  - 思路1：E(A)=E(2)=…=E(K)且∑E(x)=26 ⇒ E(K)=2
  - 思路2：P(x)=26/52=1/2 ⇒ $E(K)=E(K_{黑})+E(K_{红})+E(K_{梅})+E(K_{方})=4(1/2)=2$

CS第5.4节问题10

- $E(c)=\sum X(s)P(s)=\sum cP(s)=c\sum P(s)=c$

CS第5.4节问题21

- Give an example of a random variable … with an infinite expected value …
  - $P(F^iS)=(1-p)^ip \Rightarrow E(X)=\sum(1-p)^ipX(F^iS)=\infty$
  - 例如：$X(F^iS)=(1-p)^{-i}$

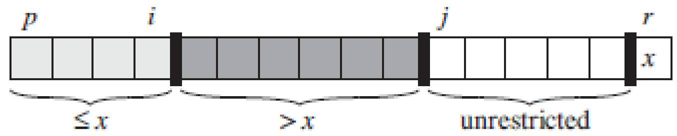
- 教材答疑和讨论
  - TC第7、8、9章

问题1：快速排序

- 你能简洁阐述快速排序的过程吗？

```
QUICKSORT(A, p, r)
if p < r
    q = PARTITION(A, p, r)
    QUICKSORT(A, p, q − 1)
    QUICKSORT(A, q + 1, r)
```

```
PARTITION(A, p, r)
x = A[r]
i = p − 1
for j = p to r − 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```



1. If $p \le k \le i$, then $A[k] \le x$.
2. If $i + 1 \le k \le j - 1$, then $A[k] > x$.
3. If $k = r$, then $A[k] = x$.

- PARTITION中的loop invariant是什么？
- 你能证明PARTITION是totally correct的吗？
- 你能证明QUICKSORT是totally correct的吗？

问题1：快速排序 (续)

- 你能描述worst-case和best-case吗？
- 它们运行时间的递归式分别是什么？

$$
\begin{aligned}
T(n) &= T(n-1) + T(0) + \Theta(n) \\
&= T(n-1) + \Theta(n) .
\end{aligned}
$$

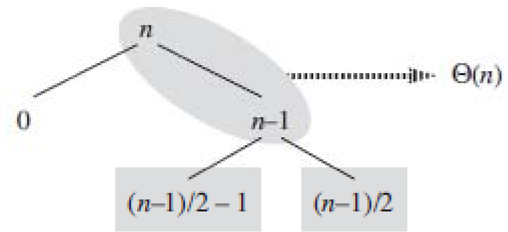
$$T(n) = 2T(n/2) + \Theta(n)$$

- 你能画出它们的recursion tree吗？

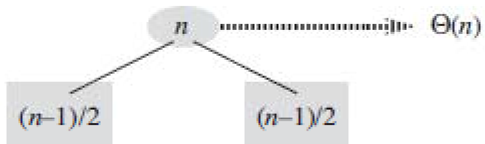
```
QUICKSORT(A, p, r)
if p < r
    q = PARTITION(A, p, r)
    QUICKSORT(A, p, q - 1)
    QUICKSORT(A, q + 1, r)

PARTITION(A, p, r)
x = A[r]
i = p - 1
for j = p to r - 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```

- 你能借助这个例子猜测average case的运行时间吗？



(a)



(b)

问题1：快速排序 (续)

- RANDOMIZED-QUICKSORT与QUICKSORT有什么不同？

```
RANDOMIZED-QUICKSORT(A, p, r)
if p < r
    q = RANDOMIZED-PARTITION(A, p, r)
    RANDOMIZED-QUICKSORT(A, p, q - 1)
    RANDOMIZED-QUICKSORT(A, q + 1, r)
```

```
RANDOMIZED-PARTITION(A, p, r)
i = RANDOM(p, r)
exchange A[r] with A[i]
return PARTITION(A, p, r)
```

- 这种改变有什么意义？
  - In exploring the average-case behavior of quicksort, we have made an assumption that all permutations of the input numbers are equally likely. In an engineering situation, however, we cannot always expect this assumption to hold.

问题1：快速排序 (续)

- RANDOMIZED-QUICKSORT的运行时间主要耗费在哪个步骤上？
- 为什么每对元素最多比较1次？
- 你能解释以下计算过程吗？

$$
\begin{aligned}
\Pr\{z_i \text{ is compared to } z_j\} &= \Pr\{z_i \text{ or } z_j \text{ is first pivot chosen from } Z_{ij}\} \\
&= \Pr\{z_i \text{ is first pivot chosen from } Z_{ij}\} \\
&\quad + \Pr\{z_j \text{ is first pivot chosen from } Z_{ij}\} \\
&= \frac{1}{j-i+1} + \frac{1}{j-i+1} \\
&= \frac{2}{j-i+1} .
\end{aligned}
$$

- 然后如何计算expected running-time？

$$
\begin{aligned}
\mathrm{E}[X] &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1} \\
&= \sum_{i=1}^{n-1} \sum_{k=1}^{n-i} \frac{2}{k+1} \\
&< \sum_{i=1}^{n-1} \sum_{k=1}^{n} \frac{2}{k} \\
&= \sum_{i=1}^{n-1} O(\lg n) \\
&= O(n \lg n) .
\end{aligned}
$$

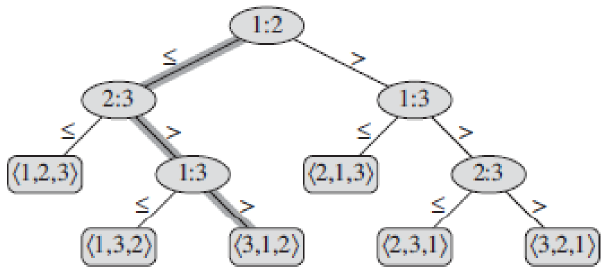
```
RANDOMIZED-QUICKSORT(A, p, r)
if p < r
    q = RANDOMIZED-PARTITION(A, p, r)
    RANDOMIZED-QUICKSORT(A, p, q − 1)
    RANDOMIZED-QUICKSORT(A, q + 1, r)
```

```
RANDOMIZED-PARTITION(A, p, r)
i = RANDOM(p, r)
exchange A[r] with A[i]
return PARTITION(A, p, r)
```

```
PARTITION(A, p, r)
x = A[r]
i = p − 1
for j = p to r − 1
    if A[j] ≤ x
        i = i + 1
        exchange A[i] with A[j]
exchange A[i + 1] with A[r]
return i + 1
```

问题2：线性时间排序算法

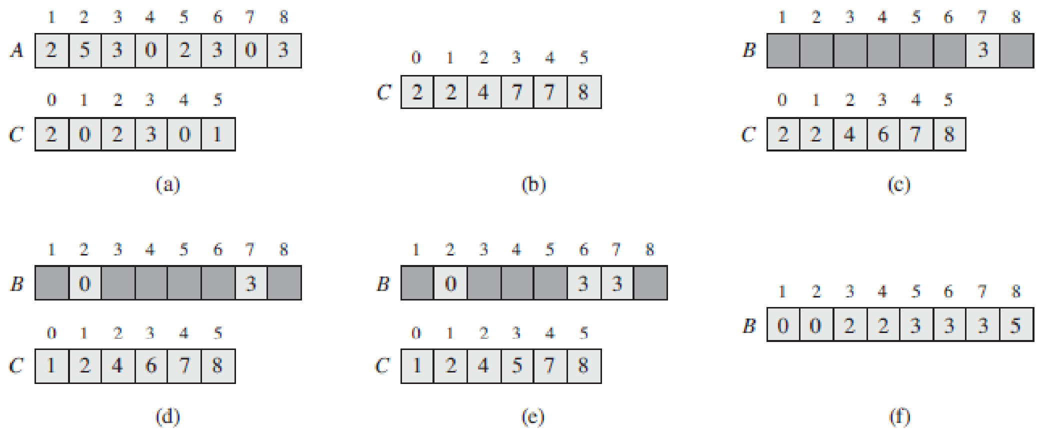
- 什么叫做comparison sorts？
  - The sorted order they determine is based only on comparisons between the input elements.
- 你是怎么理解decision tree的？它与comparison sorts的运行时间有什么关系？
  - 它有多少个叶子顶点？
  - 它有多少层？



问题2：线性时间排序算法 (续)

- counting sort的基本思路是什么？
- 为什么它是stable的？
- 能不能改为从左往右扫描？
- 它对输入有什么要求？它还有什么缺点？

```
COUNTING-SORT(A, B, k)
let C[0..k] be a new array
for i = 0 to k
    C[i] = 0
for j = 1 to A.length
    C[A[j]] = C[A[j]] + 1
// C[i] now contains the number of elements equal to i.
for i = 1 to k
    C[i] = C[i] + C[i − 1]
// C[i] now contains the number of elements less than or equal to i.
for j = A.length downto 1
    B[C[A[j]]] = A[j]
    C[A[j]] = C[A[j]] − 1
```



问题2：线性时间排序算法 (续)

- radix sort的基本思路是什么？

```
RADIX-SORT(A, d)
for i = 1 to d
    use a stable sort to sort array A on digit i
```

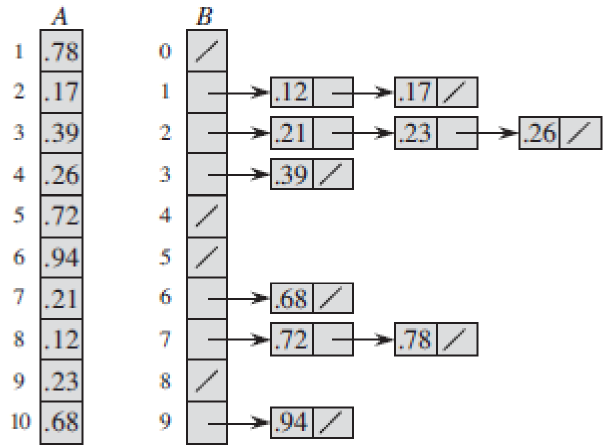
329	720	720	329
457	355	329	355
657	436	436	436
839	457	839	457
436	657	355	657
720	329	457	720
355	839	657	839

- 为什么要调用一个stable sort？
- 能不能改为从高位开始排序？
- 你怎么理解we have some flexibility in how to break each key into digits？
- 它对输入有什么要求？

问题2：线性时间排序算法 (续)

- bucket sort的基本思路是什么？

```
BUCKET-SORT(A)
let B[0..n − 1] be a new array
n = A.length
for i = 0 to n − 1
    make B[i] an empty list
for i = 1 to n
    insert A[i] into list B[⌊nA[i]⌋]
for i = 0 to n − 1
    sort list B[i] with insertion sort
concatenate the lists B[0], B[1], ..., B[n − 1] together in order
```



(a) (b)

- 它的运行时间什么时候是线性的？

$$
\begin{aligned}
\mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\
&= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}\left[O(n_i^2)\right] \quad \text{(by linearity of expectation)} \\
&= \Theta(n) + \sum_{i=0}^{n-1} O\left(\mathrm{E}\left[n_i^2\right]\right) \quad \text{(by equation (C.22))} .
\end{aligned}
$$

问题3：选择问题

- 什么是选择问题？

Input: A set A of n (distinct) numbers and an integer i, with $1 \le i \le n$.

Output: The element $x \in A$ that is larger than exactly $i - 1$ other elements of A.

- 找到最大或最小元，需要比较多少次？
- 找到最大和最小元，需要比较多少次？

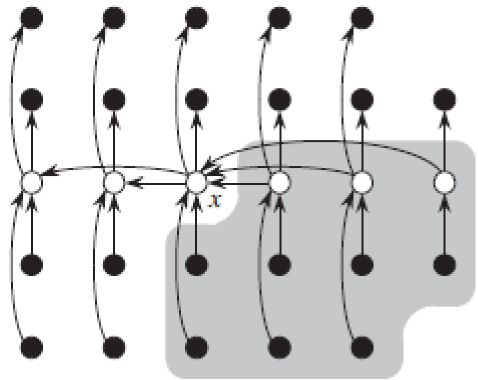
问题3：选择问题 (续)

- RANDOMIZED-SELECT的基本思路是什么？

```
RANDOMIZED-SELECT(A, p, r, i)
if p == r
    return A[p]
q = RANDOMIZED-PARTITION(A, p, r)
k = q − p + 1
if i == k          // the pivot value is the answer
    return A[q]
elseif i < k
    return RANDOMIZED-SELECT(A, p, q − 1, i)
else return RANDOMIZED-SELECT(A, q + 1, r, i − k)
```

问题3：选择问题 (续)

- SELECT算法的基本思路是什么？



- 使它成为线性算法的关键原因是什么？

$$T(n) \le \begin{cases} O(1) & \text{if } n < 140\,, \\ T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n) & \text{if } n \ge 140\,. \end{cases}$$