- 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

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

- 注意：原问题是排成一圈，所以需要减1

CS第5.2节问题15

- P272， principle of inclusion and exclusion for counting

$$N_a(\varnothing) - \sum_{k=1}^{m} (-1)^{k+1} \sum_{\substack{i_1, i_2, \ldots, i_k \\ 1 \le i_1 < i_2 < \ldots < i_k \le m}} \left| E_{i_1} \cap E_{i2} \cap \ldots \cap E_{i_k} \right|$$

$$= N_a(\varnothing) - \sum_{\substack{K \subseteq P \\ K \neq \varnothing}} (-1)^{|K|+1} N_a(K)$$

$$= \sum_{K \subseteq P} (-1)^{|K|} N_a(K)$$

- 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 → $N_e(\varnothing)$
  - $N_a(K)$ → $(m-|K|)^n$

CS第5.3节问题3、4

- 怎么证明两个event相互独立？
  - $E \cap F = \varnothing$
  - $P(E)=P(F)$

  - 定义：$P(E|F)=P(E)$
  - 定理5.5：$P(E)P(F)=P(E \cap F)$

CS第5.3节问题11

- 独立：P(E|F)=P(E)
  - $P(E\cap F)=0 \Rightarrow P(E|F)=0$ 
  - 当P(F)=0时，P(E|F)的定义是什么？

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 i-th draw independent of the event of having a king on the j-th 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 $\Rightarrow$ E(K)=2
  - 思路2：P(x)=26/52=1/2 $\Rightarrow$ $E(K)=E(K_{黑})+E(K_{红})+E(K_{梅})+E(K_{方})=4(1/2)=2$

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：$X(F^iS)=(1-p)^{-i}$

- 教材讨论
  - TC第7、8、9章

问题1：Quicksort

- 你能简述Quicksort的执行过程吗？
- PARTITION中的loop invariant是什么？
- 你能证明PARTITION是totally correct吗？
- 你能证明QUICKSORT是totally correct吗？

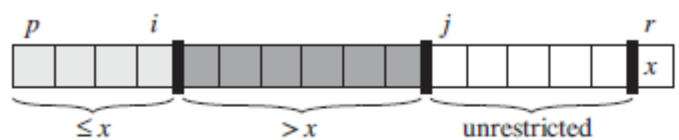
```
QUICKSORT(A, p, r)
if p < r
    q = PARTITION(A, p, r)
    QUICKSORT(A, p, q − 1)
    QUICKSORT(A, q + 1, r)
```

```
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
```

问题1：Quicksort

- 你能简述Quicksort的执行过程吗？
- PARTITION中的loop invariant是什么？
- 你能证明PARTITION是totally correct吗？
- 你能证明QUICKSORT是totally correct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 $A[k] = x$.

问题1：Quicksort (续)

- What is the running time of Quicksort when array A contains distinct elements and is sorted in decreasing order?
- What is the running time of Quicksort when all elements of array A have the same value?
- What is the best case?
  What is the worst case?

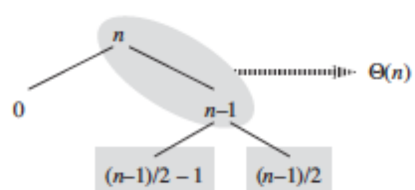
```
QUICKSORT(A, p, r)
if p < r
    q = PARTITION(A, p, r)
    QUICKSORT(A, p, q − 1)
    QUICKSORT(A, q + 1, r)

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
```

问题1：Quicksort (续)

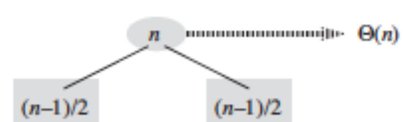
- What is the running time of Quicksort when array A contains distinct elements and is sorted in decreasing order?
- What is the running time of Quicksort when all elements of array A have the same value?

- What is the best case?
  What is the worst case?

- How about the average case?



bad



good

average?

问题1：Quicksort (续)

- RANDOMIZED-QUICKSORT与QUICKSORT有什么不同？
- 这种改变有什么意义？

- RANDOMIZED-QUICKSORT的运行时间主要耗费在什么操作？

```
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1：Quicksort (续)

- RANDOMIZED-QUICKSORT与QUICKSORT有什么不同？
- 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.
- RANDOMIZED-QUICKSORT的运行时间主要耗费在什么操作？

```
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1：Quicksort (续)

- 比较次数的计算
  - 为什么每对元素最多比较1次？
  - 你能解释以下计算过程吗？

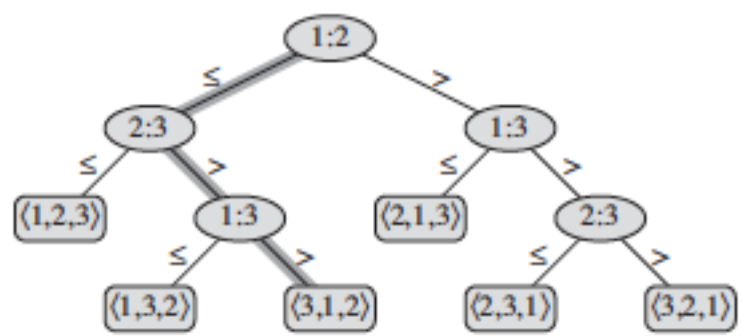
$$
\begin{aligned}
\Pr\{z_i \text{ is compared to } z_j\} &= \Pr\{z_i \text{ or } z_j \text{ is first pivot chosen from } Z_{ij}\} \\
&= \Pr\{z_i \text{ is first pivot chosen from } Z_{ij}\} \\
&\quad + \Pr\{z_j \text{ is first pivot chosen from } Z_{ij}\} \\
&= \frac{1}{j-i+1} + \frac{1}{j-i+1} \\
&=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

  - 然后如何计算expected running time？

问题2：sorting in linear time

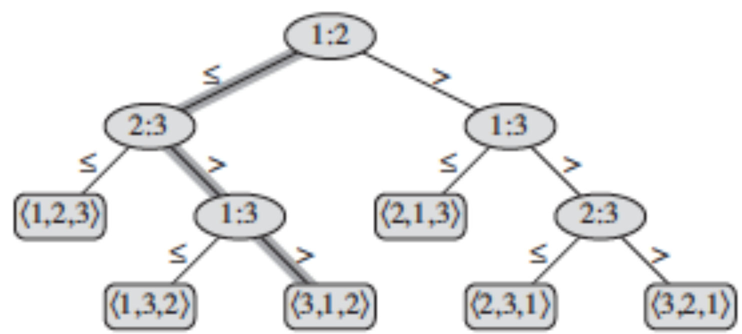
- 什么叫做comparison sorts？

- 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？
  – 它有多少层？



问题2：sorting in linear time

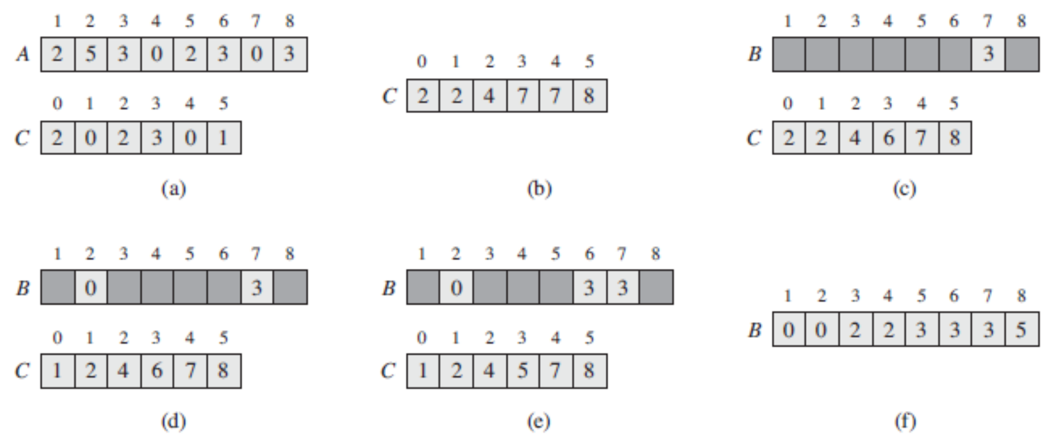
- 什么叫做comparison sorts？
  - The sorted order they determine is based only on comparisons between the input elements
- 你是怎么理解decision tree的？
  它与comparison sorts的运行时间有什么关系？
  - 它有多少个叶子？
  - 它有多少层？



问题2：sorting in linear time

- 你能简述counting sort的执行过程吗？
- 为什么它是stable的？（什么叫stable？QUICKSORT是吗？）你能不能将最后一步改为从左往右扫描，并仍保证stable？
- 它的使用有哪些局限性（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： sorting in linear time (续)

- 你能简述radix sort的执行过程吗？
- 为什么它需要调用一个stable sort？
  能不能改为从高位开始排序？
- 你如何理解
  We have some flexibility in how to break each key into digits.
- 它的使用有哪些局限性（缺点）？

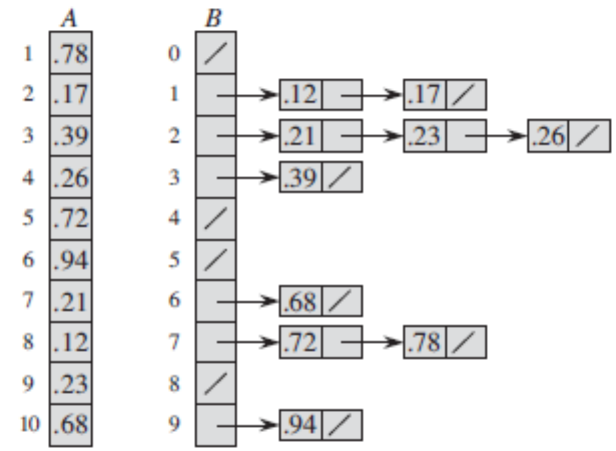
```
RADIX-SORT(A, d)
for i = 1 to d
    use a stable sort to sort array A on digit i
```

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

问题2：sorting in linear time (续)

- 你能简述bucket sort的执行过程吗？

```
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)

- 它的使用有哪些局限性（缺点）？
- 你能利用类似的思想解决以下问题吗？

We are given n points in the unit circle, $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \le 1$ for $i = 1, 2, \ldots, n$. Suppose that the points are uniformly distributed; that is, the probability of finding a point in any region of the circle is proportional to the area of that region. Design an algorithm with an average-case running time of $\Theta(n)$ to sort the n points by their distances $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin.

问题3：selection problem

- 什么是选择问题？

- 找到最大元或最小元，需要比较多少次？为什么？
- 找到最大元和最小元，需要比较多少次？为什么？

问题3：selection problem

- 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.

- 找到最大元或最小元，需要比较多少次？为什么？
- 找到最大元和最小元，需要比较多少次？为什么？

问题3：selection problem (续)

- 你能简述RANDOMIZED-SELECT的执行过程吗？
- 它的best case和worst case分别是什么？
- 你会把递归改为迭代吗？

```
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：selection problem (续)

- 你能简述SELECT的执行过程吗？
- 如果每组7个元素行不行？3个行不行？

$$3\left(\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 2\right) \geq \frac{3n}{10} - 6 .$$

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