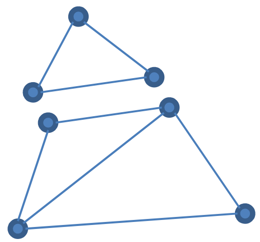
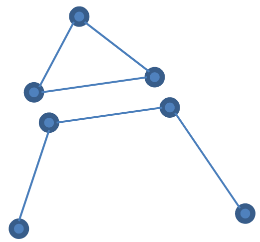
- 书面作业讲解
  - CS第4.1节问题16、17
  - CS第4.2节问题8、11、17
  - CS第4.3节问题9、13、16
  - CS第4.4节问题1、4、6
  - CS第4.5节问题8、9、10

CS第4.1节问题16

- … This version does not specify that the ears are nonadjacent. What happens if we try proving this by induction, using the same decomposition that we used in proving the Ear Lemma?





CS第4.1节问题17

- ... relationship between the number of vertices in a polygon and the number of triangles in any triangulation of that polygon ...
  - 在数学归纳法中，使用top-down而非bottom-up的表述方式，即从要证明的结论开始
  - 例如：将任意n边形拆分，而不是从任意n-1边形拼接

CS第4.2节问题8

- $T(n)=2T(n-1)+2000 \quad (n>1)$
- $T(1)=2000$

CS第4.2节问题11

- 定理4.5

CS第4.2节问题17

- 定理4.5
- 定理4.6

$$T(n) = r^n + \sum_{r=1}^{n} r^{n-i} i = r^n + r^n \sum_{r=1}^{n} i \left(\frac{1}{r}\right)^i = \ldots$$

CS第4.4节问题1

- 定理4.11（主定理的扩展形式）

CS第4.5节问题8

- 错误1
  - 欲证$T(n)\le cn^3$
  - 归纳假设$T(n/2)\le c(n/2)^3$
  - 计算$T(n)\le cn^3+n\lg n=O(n^3)+O(n^2)=O(n^3)$，得证
- 错误2
  - 欲证$T(n)\le cn^3-dn\lg n$
  - 归纳假设$T(n/2)\le c(n/2)^3-d(n/2)\lg(n/2)$
  - 计算$T(n)\le \ldots \le cn^3$，得证
- 错误3
  - 欲证$T(n)\le c_1n^3$
  - 归纳假设$T(n/2)\le c_1(n/2)^3$
  - 计算$T(n)\le c_1n^3+n\lg n\le c_1n^3+c_2n^3=c_3n^3$，得证
- 错误4
  - 欲证$T(n)\le cn^3-d$
  - 归纳假设$T(n/2)\le c(n/2)^3-d$
  - 计算$T(n)\le cn^3-d-7d+n\lg n$，只需取$d\ge n\lg n/7$，得证

CS第4.5节问题8 (续)

- 一种证法
  - 欲证$T(n) \le c(n^3-n^2)+d$
  - 归纳假设$T(n/2) \le c((n/2)^3-(n/2)^2)+d$
  - 计算$T(n) \le cn^3-2cn^2+8d+n\lg n=[c(n^3-n^2)+d]+(n\lg n+7d-cn^2)$
    $\le [c(n^3-n^2)+d]+(n^2+7d-cn^2)$
  - 只要$c \ge 7d+1$，$T(n) \le [c(n^3-n^2)+d]+(n^2+7d-(7d+1)n^2)$
    $=[c(n^3-n^2)+d]+7d(1-n^2) \le c(n^3-n^2)+d$
  - 并且，$T(1)=d \le d$也成立

- 教材答疑和讨论
  - TC第5章
  - CS第5章第6、7节

问题1：indicator random variable

- 你怎么理解indicator random variable？
- 怎么利用indicator random variable来简化期望的计算？

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} P(X_i = 1)$$

问题1：indicator random variable (续)

- 在这些问题中，indicator random variable分别是什么？
  - The expected number of times that we hire a new office assistant.
  - The expected number of pairs of people with the same birthday.

问题2：随机化

- 你怎么理解expected running time？
- 你怎么理解average-case running time？

- 它们有什么区别？
  - We discuss the average-case running time when the probability distribution is over the inputs to the algorithm, and we discuss the expected running time when the algorithm itself makes random choices.

- 你怎么理解randomized algorithm？
  - Its behavior is determined not only by its input but also by something chosen randomly (e.g. values produced by a random-number generator).

问题2：随机化 (续)

- 你觉得有哪些方法可以生成一个32-bit的（伪）随机数？
  - Computational methods (pseudo-random number generators)

```
m_w = <choose-initializer>;    /* must not be zero */
m_z = <choose-initializer>;    /* must not be zero */

uint get_random()
{
    m_z = 36969 * (m_z & 65535) + (m_z >> 16);
    m_w = 18000 * (m_w & 65535) + (m_w >> 16);
    return (m_z << 16) + m_w;  /* 32-bit result */
}
```

  - Physical methods
    - Coin flipping
    - Dice
    - Variations in the amplitude of atmospheric noise recorded with a normal radio

问题2：随机化 (续)

- 你觉得有哪些方法可以对一个数组中的元素随机排序？
  - PERMUTE-BY-SORTING(A)

```
n = A.length
let P[1..n] be a new array
for i = 1 to n
    P[i] = RANDOM(1, n^3)
sort A, using P as sort keys
```

  - RANDOMIZE-IN-PLACE(A)

```
n = A.length
for i = 1 to n
    swap A[i] with A[RANDOM(i, n)]
```

问题3： expected running time

- 你学会了哪些方式来计算E(X)？
  - $E(X)=\sum xP(X=x)$
  - $E(X)= \sum E(X_i)$
  - $E(aX+bY)=aE(X)+bE(Y)$
  - $E(X)=\sum E(X|F_i)P(F_i)$

问题3：expected running time (续)

- **Exercise 5.6-2** Let $A(1:n)$ denote the elements in positions 1 to n of the array A. A recursive description of insertion sort is that to sort $A(1:n)$, first we sort $A(1:n-1)$, and then we insert $A(n)$, by shifting the elements greater than $A(n)$ each one place to the right and then inserting the original value of $A(n)$ into the place we have opened up. If $n = 1$ we do nothing. Let $S_j(A(1:j))$ be the time needed to sort the portion of A from place 1 to place j, and let $I_j(A(1:j), b)$ be the time needed to insert the element b into a sorted list originally in the first j positions of A to give a sorted list in the first $j+1$ positions of A. Note that S_j and I_j depend on the actual array A, and not just on the value of j. Use S_j and I_j to describe the time needed to use insertion sort to sort $A(1:n)$ in terms of the time needed to sort $A(1:n-1)$. Don't forget that it is necessary to copy the element in position i of A into a variable b before moving elements of $A(1:i-1)$ to the right to make a place for it, because this moving process will write over $A(i)$. Let $T(n)$ be the expected value of S_n; that is, the expected running time of insertion sort on a list of n items. Write a recurrence for $T(n)$ in terms of $T(n-1)$ by taking expected values in the equation that corresponds to your previous description of the time needed to use insertion sort on a particular array. Solve your recurrence relation in big-Θ terms.

$$S_n(A(1:n)) = S_{n-1}(A(1:n-1)) + I_{n-1}(A(1:n-1), A(n)) + c_1$$

$$E(S_n) = E(S_{n-1}) + E(I_{n-1}) + E(c_1)$$

$$E(I_{n-1}) = \sum_{i=0}^{n-1} i\frac{1}{n} = \frac{1}{n}\sum_{i=0}^{n-1} i = \frac{1}{n}\frac{(n-1)n}{2} = \frac{n-1}{2}$$

问题3：expected running time (续)

- Slower Quicksort(A,n)

```
if (n = 1)
    return the one item in A
else
    Repeat
        p = randomElement(A)
        Let H be the set of elements greater than p; Let h = |H|
        Let L be the set of elements less than or equal to p; Let ℓ = |L|
    Until (|H| ≥ n/4) and (|L| ≥ n/4)
    A1 = QuickSort(H,h)
    A2 = QuickSort(L,ℓ)
    return the concatenation of A1 and A2
```

$$T(n) \leq E(r)bn + T(a_n n) + T((1-a_n)n)$$

问题3： expected running time (续)

- **Exercise 5.6-4** Consider an algorithm that, given a list of n numbers, prints them all out. Then it picks a random integer between 1 and 3. If the number is 1 or 2, it stops. If the number is 3 it starts again from the beginning. What is the expected running time of this algorithm?

$$T(n) = \frac{2}{3}cn + \frac{1}{3}(cn + T(n))$$

问题3：expected running time (续)

- `RandomSelect(A,i,n)`

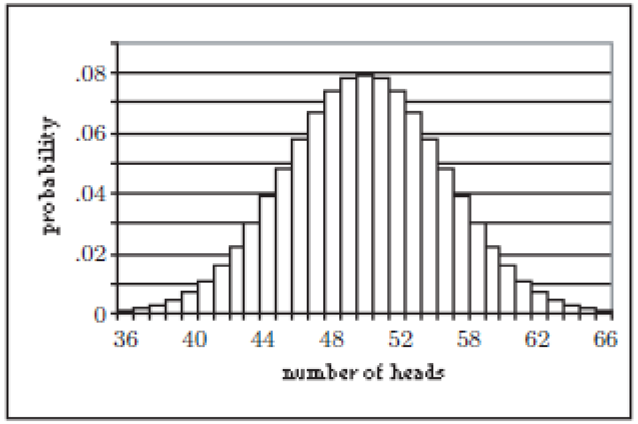
```
(selects the ith smallest element in set A, where n = |A| )
if (n = 1)
    return the one item in A
else
    p = randomElement(A)
    Let H be the set of elements greater than p
    Let L be the set of elements less than or equal to p
    If (H is empty)
        put p in H
    if (i ≤ |L|)
        Return RandomSelect(L, i, |L|)
    else
        Return RandomSelect(H, i − |L|, |H|).
```

When we choose our partition element, half the time it will be between $\frac{1}{4}n$ and $\frac{3}{4}n$. Then when we partition our set into H and L, each of these sets will have no more than $\frac{3}{4}n$ elements. The other half of the time each of H and L will have no more than n elements.

$$T(n) \le \begin{cases} \frac{1}{2}T(\frac{3}{4}n) + \frac{1}{2}T(n) + bn & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases}$$

问题4：probability distributions and variance

- 你怎么理解distribution function和它的histogram？



- 你怎么理解cumulative distribution function？
- 它有哪些性质？
- 什么情况下只能使用cumulative distribution function？

问题4： probability distributions and variance (续)

- 你怎么理解variance？
- 人们是怎么想到把variance定义成 $V(X) = E(X - E(X))^2$ 的？