Introduction to Algorithms

ZHENG Hong

Department of Computer Science and Engineering

East China University of Science and Technology

Problem-1

- Problem-1
- The 3 teachers predicted a student competition. Their predictions are as follows:
- A said: student A is the first, student B is the third.
- B said: Student C was the first and student D was the fourth.
- C said: Student D is the second, student A is the third.
- The results show that they are half right, and half wrong with no ranking .
- Try to program the input a, b, c, d their respective rankings .

Problem-2

A polygon is ***convex*** if all of its internal angles are less than 180° (and none of the edges cross each other). Figure 1 shows an example. We represent a convex polygon as an array $V[1..n]$ where each element of the array represents a vertex of the polygon in the form of a coordinate pair (x, y). We are told that $V[1]$ is the vertex with the minimum x coordinate and that the vertices $V[1..n]$ are ordered counterclockwise, as in the figure. You may also assume that the x coordinates of the vertices are all distinct, as are the y coordinates of the vertices.

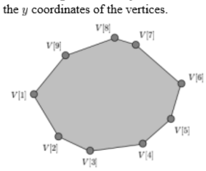

Figure 1: An example of a convex polygon represented by the array $V[1..9]$. $V[1]$ is the vertex with the minimum x-coordinate, and $V[1..9]$ are ordered counterclockwise.

(b) Give an algorithm to find the vertex with the maximum x coordinate in $O(\lg n)$ time.

(c) Give an algorithm to find the vertex with the maximum y coordinate in $O(\lg n)$ time.

A convex polygon

- A convex polygon is a simple polygon (not self-intersecting) in which no line segment between two points on the boundary ever goes outside the polygon. Equivalently, it is a simple polygon whose interior is a convex set.[1]
- In a convex polygon, all interior angles are less than or equal to 180 degrees, while in a strictly convex polygon all interior angles are strictly less than 180 degrees.

The properties of a simple polygon

- The following properties of a simple polygon are all equivalent to convexity:

① Every internal angle is less than or equal to 180 degrees.

② Every point on every line segment between two points inside or on the boundary of the polygon remains inside or on the boundary.

③ The polygon is entirely contained in a closed half-plane defined by each of its edges.

④ For each edge, the interior points are all on the same side of the line that the edge defines.

⑤ The angle at each vertex contains all other vertices in its edges and interior.

⑥ The polygon is the convex hull of its edges.

Problem-3：merge sorted

- Given two sorted integer arrays nums1 and nums2, merge nums2 into nums1 as one sorted array.
- Note:
- The number of elements initialized in nums1 and nums2 are m and n respectively.
- You may assume that nums1 has enough space (size that is greater or equal to m + n) to hold additional elements from nums2.
- Example:
- Input:
- nums1 = [1,3,4,0,0,0], m = 3
- nums2 = [2,5,7],   n = 3

- Output: [1,2,3,4,5,7]

Problem-4： Pattern Matching

- Mr. Li has a problem: he is absolutely sure that Bart Simpson has plagiarized some text on a recent book report. One of Bart’s sentences sounds oddly familiar, but Skinner can’t quite figure out where it came from. Please help him out.

Problem-5： binary search tree

- If we insert a set of n items into a binary search tree using TREE-INSERT, the resulting tree may be horribly unbalanced. As we saw in class, however, we expect randomly built binary search trees to be balanced. (Precisely, a randomly built binary search tree has expected height O(lg n).) Therefore, if we want to build an expected balanced tree for a fixed set of items, we could randomly permute the items and then insert them in that order into the tree. What if we do not have all the items at once? If we receive the items one at a time, can we still randomly build a binary search tree out of them?

Problem-6： counts

- Question: In a ordered array, some elements are duplicated.
- Write a function, the number of times we return to this number for a given number.
- For example, input data 1 2 2 3 3 3 5 5, if input 2, return 2, because 2 appears in the array 2 times.

Problem-7： Find a number

- Find the first number greater than (or less) specified. It's a bit hard to understand. Let's give an example. (For example, the ascending array 1 4 5 6 8 9, if the input is 7, we should return 8, because 8 is the first number greater than the input: 7)

Problem-8： Find paths

- Find all paths for a certain value in a binary tree.
- Topic: enter an integer number and a binary tree.
- From the root node of the tree, we have to go down until all the nodes passed through the leaf node form a path.
- Print all the paths that are equal to the input number.
-
- For example: input “22”

```
      10
     /  \
    5    12
   / \
  4   7
```

- Output two paths : “10, 12”; “10, 5, 7”

Problem-8： Maximum sum of sub-arrays

- Maximum sum of sub-arrays
- Input an array of integers, positive and negative numbers in the array.
- One or more integers in an array form a subarray, and each subarray has a sum.
- Seek the maximum of the sum of all subarrays. The required time complexity is O (n).
- For example, the input array is 1, -2, 3, 10, -4, 7, 2, -5, and the largest subarray is 3, 10, -4, 7, 2, so the output is the subarray and 18.

Problem-9： Find the smallest k elements

- Find the smallest k elements
- Topic: input n integer, output the smallest K.
- For example, if you import 1, 2, 3, 4, 5, 6, 7 and 8（8 numbers）, the smallest 4 numbers are 1, 2, 3 and 4.

Problem-10： Find the smallest k elements

- Input an English sentence and reverse the order of the words in the sentence, but the order of the characters in the word remains unchanged.
- The words in the sentence are separated by spaces. For simplicity, punctuation is handled just like ordinary letters.
- For example, input “you are a student.", then output "student. a are you".

Problem-11： dynamic programing-N matrix multiplication

- Given n matrix $\{A_1, A_2, \ldots, A_n\}$
- Where A_i and A_{i+1} are multiplicative.
- Examine the continuous product $\{A_1, A_2, \ldots, A_n\}$ of these n matrices.
- How to determine the calculation order of matrix continuous product, so that the order of calculation matrix continuous product requires the least number of multiplications.

Problem-12： Jumping step problem

- If there are n steps, if you can jump 1 step once, you can also jump 2 steps once.
- The total hop method is analyzed and the time complexity of the algorithm is analyzed.

Problem-13

- The number of 1 in the binary representation of an integer.
- Problem : input an integer and find the number of binary expressions in the integer number 1.
- For example, input 10, because its binary representation is “1010”, and it has two “1”, so the output “2”.

Problem-14： Identical coins

- There are eight identical coins, one of which is counterfeit, and it is known that counterfeit coins weigh different from real ones, but it is not known whether counterfeit coins are lighter or heavier than real ones. T
- An efficient algorithm is designed to detect the counterfeit coin.

Problem-15:

- Input an integer n and count the number of occurrences of 1 out of the decimal representation of N integers from 1 to n.
- For example, enter 13, and the numbers 1, 10, 11, 12and 13, 1 in the integers from 1 to 13 appear six times.

Problem-16: GreedSox

- GreedSox, a popular major-league baseball team, is interested in one thing: making money. They have hired you as a consultant to help boost their group ticket sales. They have noticed the following problem.
- When a group wants to see a ballgame, all members of the group need seats (in the bleacher section), or they go away.
- Since partial groups can't be seated, the bleachers are often not full. There is still space available, but not enough space for the entire group. In this case, the group cannot be seated, losing money for the GreedSox.

Problem-16

- The GreedSox want your recommendation on a new seating policy. Instead of seating people firstcome/first-serve, the GreedSox decide to seat large groups first, followed by smaller groups, and finally singles (i.e., groups of 1). You are given a set of groups, G[1 ..m] = [g1,g2,...,gm], where gi is a number representing the size of the group. Assume that the bleachers seat n people. Consider the following greedy seating algorithm, where the function ADMIT(i) admits group i, and REJECT(i) sends away group i.

Problem-17:

- Enter a string ,and ouput all strings that can be arranged all the characters in the string .
- For example, the input string "a b c," .then the ouput all strings that can be arranged by the characters a, b, and C.
- outputs strings :" abc, acb, bac,bca,cab, and cba ".

Problem-18: Find a different ball

- there are 12 balls, they have the same shape, but the quality of one ball is different from the other 11 balls. Question:Using a balance, ask how to find the ball with three tests and find out if it is lighter or heavier than the others.

Problem-19: Machine scheduling

- The existing n task and unlimited number of machines.
- Each task can be processed on one machine.
- The start time of each task is S_i, and the completion time is f_i, $S_i < f_i$. $[s_i, f_i]$ to handle the time range of task I.
- Two tasks I and j refer to the overlap of the time range of two tasks, not the overlap of the starting point or the end point of I and J.
- For example, the interval [1, 4] overlaps with the interval [2, 4] and does not overlap with the interval [4, 7].

Problem-19: Machine scheduling

- A feasible task assignment is that no two overlapping tasks are allocated to the same machine in the allocation. Therefore, in a feasible allocation, each machine will handle at most one task at any time. The optimal allocation is the feasible allocation scheme for the least machine.
- Suppose there are 7 tasks(n=7), and the task labels are a to g. Their start time and completion time are shown in figure 13-1a. If the task a to M1 machine, B machine to M2 task, G task. And, to machine M7, the allocation is feasible allocation, a total of seven machines. But it is not optimal, because other allocation scheme can make use of the machine number less, for example, the task can be a, b, d assigned to the same machine, the number of machines dropped to five units.

Problem-19: Machine scheduling

任务	a	b	c	d	e	f	g
开始(si)	0	3	4	9	7	1	6
完成(fi)	2	7	7	11	10	5	8

Problem-20: Performce

- Flute's Phlharmoniker Orchestra is scheduled to go to Harp city for a major performance in 2018. The Orchestra conductor, L.Y.M., is scheduled to make a tour around some small cities before arriving in Harp.
- For the next few days, musicians will fly from one city to another on a daily flight before they arrive. Destination Harp city (the orchestra can perform many times in the same city).
- Because the cost of the route and the number of flights are changing every day, there is a circular flight schedule between the city and the city. Each time, each direction, the cycle of the flight schedule may be different.

Problem-21:a **partiion** of a positive integer *n*

- a **partition** of a positive integer *n*, also called an **integer partition**, is a way of writing *n* as a sum of positive integers. Two sums that differ only in the order of their summands are considered the same partition. (If order matters, the sum becomes a composition.)
- such as:
- 5=5
- 5=4+1
- 5=3+2
- 5=3+1+1
- 5=2+1+1+1
- 5=1+1+1+1+1

Problem-22: Continuous postage problem

- Suppose there are n stamps of different denominations and and only m stamps are allowed on each envelope at most.
- For a given value of *n* and *m*, the best design of the face value of stamps is required.
- The maximum continuous postage range with an increment of 1 from postage 1 can be posted on an envelope.
- For example, when n = 2 and M = 3, if the denominations are 1 and 4, each postage value between L-6 can be obtained (8, 9, and 12 of course).

Problem-23: Multiplication algorithm

- **The** algorithm is used to calculate the multiplication of large numbers .
- Given two non-negative integers num1 and num2 represented as strings, return the product of num1 and num2.
- This example uses *long multiplication* to multiply 23,958,233 (multiplicand) by 5,830 (multiplier) and arrives at 139,676,498,390 for the result (product).

Problem-24: Complex multiplication algorithm

- **Complex multiplication normally involves four multiplications and two additions.**

$$(a+bi)(c+di) = (ac-bd) + (bc+ad)i.$$

- Or

$\times$	a	bi
c	ac	bci
di	adi	$-bd$

- **But there is a way of reducing the number of multiplications to three.**

The product $(a + bi) \cdot (c + di)$ can be calculated in the following way.
$k_1 = c \cdot (a + b)$
$k_2 = a \cdot (d - c)$
$k_3 = b \cdot (c + d)$
Real part $= k_1 - k_3$
Imaginary part $= k_1 + k_2$.

Problem: 8216547*96785=?

Problem-25: Big Integer Addition

- Write an algorithm to calculus the sum of two non negative integers that do not exceed 200 bits.
- Given two non-negative integers num1 and num2 represented as string, return the sum of num1 and num2.
- Problem: 1223456111111+654321222222=?

Problem-26: The shortest-path problem

- In graph theory, the shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.
- For example: The figure represents the road network between cities, and the number on the line segment represents the cost. The problem of finding the shortest cost between two cities.