高级算法分析与设计 Assignment 1

1. Arrange the following functions in ascending asymptotic order of growth rate:

(a) $f_1(n) = n^{2.024} + 2024^{100}n^2$，$f_2(n) = 2024^{\log n + \log\log n}$，$f_3(n) = \sqrt{n^{3.5}}$，

$f_4(n) = 2^{2n}$，$f_5(n) = 3^n$；

(b) $f_1(n) = n^{\log^2 n}$，$f_2(n) = 2^{\log n + \log\log n}$，$f_3(n) = \log^n \log^2 n$，$f_4(n) = n^{\sqrt{n}\log n}$.

方法：分析各个函数的渐进确界（Big-Θ notation），写出化简过程，得步骤分。

(1) 直接化简成多项式或者指数形式比较；

(2) 同时取 log 后化简比较；

(3) 两个函数相除后取极限比较。

2. Solve the following recurrences:

1、$f(n) = 12f(\frac{n}{8}) + O(n\log n).$

2、$f(n) = 3f(n-3) + O(n).$

3、$f(n) = 3f(\frac{n}{2}) + O(n^2).$

方法：先尝试主定理，后使用另外两种方法解题，写出计算过程，得步骤分。

(1) 主定理；

(2) 递归树或代数拆分；

(3) 先猜后代入验证。

3. Given currency denominations: 1,5,10,25,100, devise a method to pay amount x to customer using fewest number of coins. How about the case that the currency denominations are 1, 5, 7, 35, 70?

Solution：

(1) [1,5,10,25,100]可以使用贪心算法。

(2)

考虑动态规划：令$D[i]$表示第i种硬币的面值，那么我们有$D=[1,5,10,25,100]$或$D=[1,5,7,35,70]$。

令$f[i][n]$表示用前i种硬币凑出n所使用的最少硬币数量，则转移方程为：

$$f[i][n]=\begin{cases} & (\qquad\qquad) \\ & (\qquad\qquad) \end{cases}$$

$f[5][x]$即为答案。

方法：

(1) 贪心算法。（注意甄别能否使用）；

(2) 动态规划。

解题步骤：

(1) 使用 xx 算法；

(2) 用文字描述算法，不可写代码或伪代码（以动态规划为例）：

① 函数定义；

② 写出转移方程，注意不要漏掉边界条件；

(3) 如果有给出具体例子，需要按自底向上运行算法，画出取值表，给出答案。

4. Please using dynamic programming to solve the following knapsack problem. We are given 7 items and a knapsack. Each item i has weight of $w_i > 0$ kilograms and value of $v_i > 0$ dollars (given in table 1). The capacity of the knapsack is 14 kilograms. Then how to fill the knapsack to maximize the total value?

Items	Weight	Value
1	3	2
2	4	3
3	3	4
4	2	2
5	7	6
6	6	4

7	6	5

Table 1

Solution：

令$w[i]$表示第i个item的weight，$v[i]$表示第i个的value。

令$f[i][j]$表示考虑了前i个item选或不选，背包容量为j时所能获得的最大value，则有转移方程：

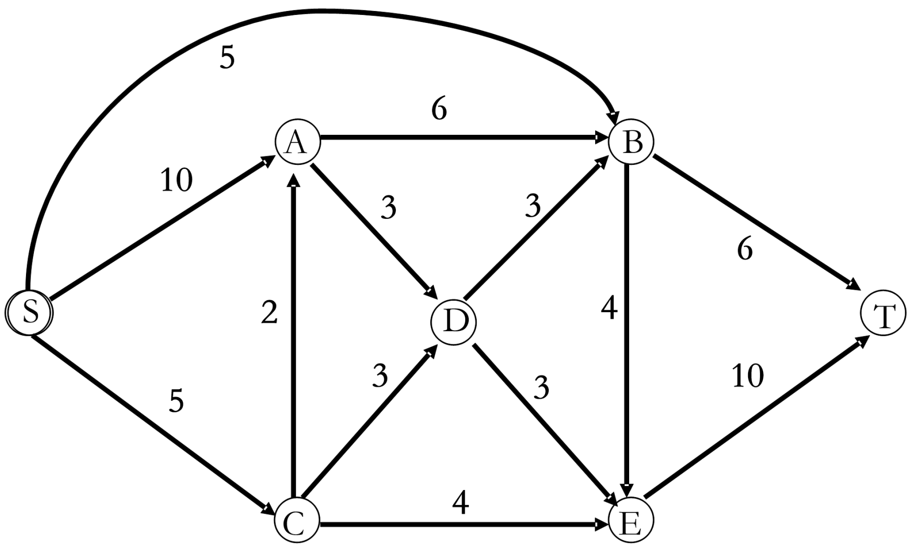
$$f[i][j] = \begin{cases} \qquad\qquad\qquad\qquad (\qquad\qquad\qquad) \\ \qquad\qquad\qquad\qquad\quad (\qquad\qquad) \\ \qquad\qquad\qquad (\qquad\qquad\qquad\quad) \end{cases}$$

下表为$f[i][j] (0 \le i \le 7, 0 \le j \le 14)$的取值表:

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0															
1															
2															
3															
4															
5															
6															
7															

因此答案为$f[7][14] =$

5. Compute a maximum flow from S to T in the following graph.



Solution: :

第一轮增广：

第二轮增广：

第三轮增广：

第四轮增广：

第五轮增广：

无法继续找到从S到T的增广路，因此从S到T的最大流为

解题步骤：

(1) 按照最大流算法给出每一轮的增广路径和剩余图（体现过程，不能省略直接写答案）；

(2) 给出最终流量图并计算出最大流。

6. You have a box of identical eggs and you need to find out the strength of these eggs. The strength of an egg is measured by an integer i from 1 to 9, which corresponds to a height h_i (for any 1 <= i < 9, we have $h_i < h_{i+1}$) that the egg will not be broken when it is dropped at or below that height. The test is done by dropping an egg at a height h_i. If it is not broken, then pick up the egg and drop it again from a new height $h_j > h_i$; otherwise use a new egg and drop it at a lower height. Repeat the process until the strength of the eggs is determined. Design a strategy that uses as few drops as possible, under the condition that you can break at most two eggs, to determine the strength of the eggs.
How about under the condition that you can break at most three eggs?

Solution：动态规划

考虑有n个鸡蛋，从m个高度测出结果至少需要丢几次。令$f[i][j]$表示还剩下i个鸡蛋，需要从j个高度中测出结果的最少次数。

若只有1个鸡蛋，那么我们只能从第一个高度开始往上逐个尝试，直至鸡蛋碎，因此1个鸡蛋测m个高度最坏情况下需要丢m次。

若有$i(i \geq 1)$个鸡蛋，假设先用1个鸡蛋从第$k(1 \leq k \leq j)$个高度丢下，若该鸡蛋碎了，那么我们可以确定结果在1到$k-1$这$k-1$个高度中，我们还有$i-1$个鸡蛋；否则若鸡蛋没碎，我们可以确定结果在$k+1$到j这$j-k$个高度中，我们还有i个鸡蛋。因此有转移式：

$$f[i][j] = \begin{cases} & (\quad) \\ & (\quad) \end{cases}$$

i/j	1	2	3	4	5	6	7	8	9	10
1										
2										
3										