CS4311
Design and Analysis of Algorithms

Tutorial: Hints on Assignment 4
Speaker: 古宗翰 (Wisely)
Outline

• Description + Hints for Assignment 4
Question 1

- We have a lot of sushi in a table.
- They are arranged into a line.

Each sushi has a quality value (higher if it tastes better)
The quality values of sushi are distinct.
Question 1

• Find a way to take sushi such that
  (1) total quality value is maximized, and
  (2) the quality value of sushi taken is increasing
    i.e. when you take a sushi A, you must next take a
    sushi B better than A.

• Your algorithm should run in $O(n^2)$ time

  Hint: Suppose you know the best way to take the first
  k sushi, for each $k = 1, 2, \ldots, n-1$.
  Can you tell the best way to take the n sushi?
The following are some possible ways to take the sushi.
(1) 5, 8, 10, 11.
(2) 5, 8, 13, 20.
(3) 5, 8, 10, 20.
(4) 8, 13.
(5) ...
It turns out (2) is the maximum among all possible ways.
The following are some possible ways to take the sushi.
(1) 3, 4, 50.
(2) 3, 4, 5, 6, 10.
(3) 3, 4, 5, 6, 8.
(4) …
It turns out (1) is the maximum among all possible ways.
Question 2

There is a stair and there is a coupon in each stage of the stair. Each coupon is associated with a value.

You need to climb to the top in at most L steps. Since you are in a hurry, you need to climb 1, 2 or 3 stages (upwards) in each step. When at a stage, you could collect the coupon on that stage.
Question 2

• Find a way to climb the stair and pick up the maximum value of the coupons in O(n^2) time.

• Hint: Suppose you know the best way to climb to stage k, using exactly X steps, for each k = 1, 2, …, n-1. Do you know the best way to climb to stage k+1, using exactly X+1 steps?
Example

<table>
<thead>
<tr>
<th>s0</th>
<th>s1</th>
<th>s2</th>
<th>s3</th>
<th>s4</th>
<th>s5</th>
<th>s6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$30</td>
<td>$40</td>
<td>$20</td>
<td>$10</td>
<td>$30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You want to climb to stage 6 in at most 3 steps.

Then the answer should be s2, s5, s6.
Example

<table>
<thead>
<tr>
<th></th>
<th>s0</th>
<th>s1</th>
<th>s2</th>
<th>s3</th>
<th>s4</th>
<th>s5</th>
<th>s6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>$30</td>
<td>$40</td>
<td>$80</td>
<td>$10</td>
<td>$30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You want to climb to stage 6 in at most 3 steps.

Then the answer should be s2, s3, s6.
There is a city which looks like a tree, with buildings = nodes, roads = edges.

We want to open stinky-tofu stores in the city.
Each building can open a store, and it has a value representing # customers who will visit the building.

Each store cannot be adjacent to each other (i.e., stores cannot be located on the same edge).
Question 3

Target: Open the stores so as to maximize the total # customers

Your algorithm should run in $O(n)$ time.

Hint: Bottom-up approach.
In this tree, we should open the stinky-tofu stores in the buildings with red numbers.
In this tree, we should open the stinky-tofu stores in the buildings with red numbers.
What shall I eat?
Question 4

- Each dish has a value and its volume, where each volume is an integer.
- Your stomach has a capacity of $V$ units, where $V$ is an integer.
- Find a way to eat dishes with the maximum total value such that your stomach could hold.

- Your algorithm should run in $O(nV)$ time.
- Hint: optimal substructure.
Example

<table>
<thead>
<tr>
<th>Dish</th>
<th>Orange</th>
<th>Beef</th>
<th>Vegetable</th>
<th>Cake</th>
<th>Pudding</th>
<th>Apple</th>
<th>Tofu</th>
<th>Egg</th>
<th>Chicken</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$</td>
<td>30</td>
<td>100</td>
<td>10</td>
<td>40</td>
<td>15</td>
<td>35</td>
<td>10</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>volume</td>
<td>15</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>30</td>
<td>8</td>
<td>60</td>
</tr>
</tbody>
</table>

Assume V=80

The answer should be beef + cake + orange + apple.
Thank You