**Problem Session Week 13 Wednesday 23 May**

# Question 1

Given the following instance of the 0/1 Knapsack problem

|  |  |  |
| --- | --- | --- |
| item | weight | value |
| 1  2  3 | 2  1  3 | $12  $10  $20 |

The Knapsack Capacity W=3

Let V[i, j] be the value of the most valuable subset of the first i items that fit into the Knapsack of capacity j. Then V[i, j] can be recursively defined as following:

0 if *i*  0 *or j*  0

*V* [*i*, *j*]  max{*V* [*i*  1, *j*], *v*  *V* [*i*  1, *j*  *w* ]} if *j*  *w*  0



*i i i*

 *V* [*i*  1, *j*]



if *j*  *wi*  0

1. Using dynamic programming, complete the following table.

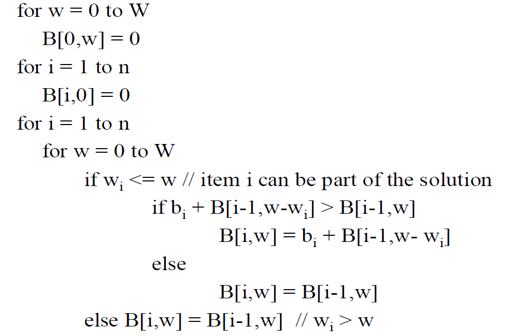
capacity j

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item *i* | 0 | 1 | 2 | 3 |
| 0 | 0 | 0 | 0 | 0 |
| *w*1=2, *v*1=12 1 | 0 | 0 | 12 | 12 |
| *w*2=1, *v*2=10 2 | 0 | 10 | 12 | 22 |
| *w*3=3, *v*3=20 3 | 0 | 10 | 12 | 22 |

1. Give an optimal subset of the instance based on the table.

The optimal subset is item 1 and 2.

1. Based on the recurrence relation given above, write a pseudocode of the bottom-up dynamic programming algorithm for the knapsack problem.

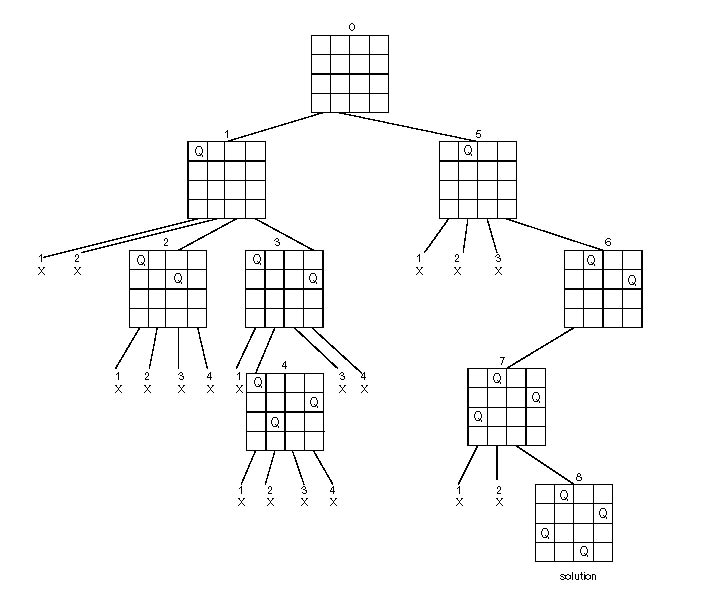


1. What is the value of the most valuable subset if the capacity of the knapsack is 2?

12

# Question 2

## Continue the backtracking search for a solution to the four-queens problem, which was given in week 11’s lecture, to find the second solution to the problem. Explain how the board’s symmetry can be used to find the second solution to the four-queens problem.



**Answer:** Since the board is symmetric, if the first queen is at 1st row and 4th column, that is the same situation as node 1. Similarly, if the first queen is at 1st row and 3rd column, that is the same situation as node 2. Therefore, the second solution is an axisymmetric shape of the first solution, which is shown below.

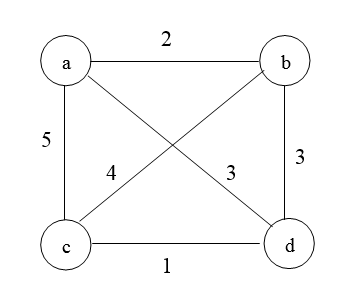
|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Q |  |
| Q |  |  |  |
|  |  |  | Q |
|  | Q |  |  |

**Question 3**

1. Is the following graph a Euclidean graph?

No, it isn’t.

1. Apply Twice-Around-the-Tree algorithm to solve the travelling salesman problem for the following graph.



|  |  |
| --- | --- |
| (c, d) | 1 |
| (a, b) | 2 |
| (b, d) | 3 |
| (a, d) | 3 |
| (c, b) | 4 |
| (a, c) | 5 |

The following graph is MST.



DFS walk: abdcdba

Eliminate repeated nodes: abdca, the length is 11.

1. What is the best solution?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a | b | c | d |
| a | 0 | 2 | 5 | 3 |
| b | 2 | 0 | 4 | 3 |
| c | 5 | 4 | 0 | 1 |
| d | 3 | 3 | 1 | 0 |

Lb=(2+3+2+3+1+4+1+3)=9.5

Thus the lowest bound is 10.



1. What is the accuracy ratio?

r(sa) = f(sa)/f(s\*)=11/10=1.1