CSC373 A2 solutions

Instructor: Milad Eftekhar

Summer 2013

This document contains the key ideas to solve A2 questions (not the complete solutions). Note that many details may be ignored in several places.

Q1.

Define the semantic array as follows:

$$V[j,c] = \text{The maximum value obtainable using items } \{I_1, \cdots, I_j\} \text{ and a knapsack size bound } c$$

To calculate $V[j,c]$ we have several options for item I_j: we may choose 0, 1, 3, ... instances of I_j. We calculate the maximum obtainable value of all of these options and select the maximum. Thus:

$$V[j,c] = \max_k \{V[j-1, c-k\times w_j] + k \times v_j\} \text{ for } k \in \{0,1,3,\cdots,h_j\}$$

$$\text{where } h_j \text{ is the largest odd number that is} \leq \lfloor c/w_j \rfloor$$

Also set $V[j,0] = 0$ and $V[0,c] = 0$. The solution is $V[n,C]$.

Complexity: Calculate $O(nC)$ entries of V. Each entry takes $O(\frac{C}{\min w_j}) = O(n)$. Thus the run time complexity is $O(nCn) = O(n^4)$.

Q2.

$Possible[i,j,x]$: Is it possible to parenthesize the substring from index i to index j ($s_i s_{i+1} \cdots s_j$) such that the result is x where x is a, b, or c?

"Possible" is a boolean 3-dimensional array. The solution is $Possible[1,n,a]$.

According to the table in question, we know

$$\begin{cases} a = ac = bc = ca \\ b = aa = ab = bb \\ c = ba = cb = cc \end{cases}$$

Thus:

$$Possible[i,j,a] = \bigvee_k \begin{pmatrix} Possible[i,k,a] \wedge Possible[k+1,j,c] \\ Possible[i,k,b] \wedge Possible[k+1,j,c] \\ Possible[i,k,c] \wedge Possible[k+1,j,a] \end{pmatrix} \text{ for all } i \leq k < j$$

In the aforementioned equation, $Possible[i, j, a]$ is $True$ if at least one of the above $3(j - i)$ terms is $True$.

Similarly find the relation for $Possible[i, j, b]$ and $Possible[i, j, c]$.
Also initialize the values:

$$Possible[i, i, x] = \begin{cases} True & \text{if } s_i = x \\ False & \text{otherwise} \end{cases} \quad \text{for all } 1 \leq i \leq n \text{ and } x \in \{a, b, c\}$$

The complexity is $O(n^3)$.
Q3.
$Possible[v', k']$: Is it possible to make change for v' using at most k' coins, of denominations $x_1, \cdots, x_n$?
$Possible[v', k'] = \bigvee_i \big(Possible[v' - x_i, k' - 1]\big)$ for all $1 \leq i \leq n$.
Initialize $Possible[0, k'] = True$ and $Possible[v', 0] = False$ for any $0 \leq k' \leq k$ and $0 < v' \leq v$.
The complexity of this algorithm is $O(nkv)$.
Q4.
Let $P_{ij} = \sum_{k=i}^{j} p_k$.
$cost[i, j]$ = the cost of the opitmal binary search tree for words $w_i \cdots w_j$.
Any word w_k from w_i to w_j can be the root of the optimal subtree for words $w_i \cdots w_j$. In that case, the words $w_i \cdots w_{k-1}$ would form the left subtree and the words $w_{k+1} \cdots w_j$ would form the right subtree. We calculate the cost of all of these options and get the minimum.
$cost[i, j] = \min_k \big(cost[i, k - 1] + cost[k + 1, j] + P_{ij}\big)$ for all $i \leq k \leq j$. Here assume $cost[i, j] = 0$ if $i > j$. (Save the best k to create the optimal subtree. Word w_k is the root of this optimal subtree)
The lowest cost of the optimal tree for all words is $cost[1, n]$ and the complexity is $O(n^3)$.
Q5.
Max flow = 13
Min cut: $V_s = \{S, C, F\}$, $V_t = \{A, B, D, E, G, T\}$.
Q6.
Let G denote the graph in the question and f is a max flow for G.
Property: For any edge e of G, if e is a bottleneck edge then $f(e) = c(e)$.
Identify a max flow f. Create another graph G' similar to G (the same nodes and edges) with new edges' weights. For any edge e in G: if $f(e) = c(e)$, set the weight of the corresponding edge in G' to 1. For any edge e in G: if $f(e) < c(e)$, set the weight of the corresponding edge in G' to 0. Find all paths with a weight of 1 from s to t in G' (run DFS). The edges with weight 1 on these paths are the bottleneck edges. (why? If you increase the capacity of any of these edges since the edge is on a path from s to t with all other edges having unused capacity, we can push more flow on the path. Any bottleneck edge has this property.)
Q7.
Create a bipartite graph $G = (L, R, E)$. There is a node in L for each injured individual. There

is a node in R for each hospital. There is a directed edge from node $u \in L$ to node $v \in R$ if the location of individual u is within a 1km distance of the hospital v. Add a node s and t to this graph. Connect s to each node in L (directed) with a capacity 1. Connect each node in R to t (directed) with capacity $\lceil n/k \rceil$. The solution to the problem is YES iff the max flow in this graph is n.