Exercises for Unit 23

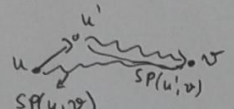
Problem 1.

We use the subpath property:

Consider u' the first node on the shortest path from u to v. Then the shortest path from u' to v is part of the shortest path from u to v. So we define:

$a(i,j) = k$, where k is the index of the first vertex on the S.P. from v_i to v_j. Then $SP(v_i, v_j) = (i,k) \cup SP(v_k, v_j)$.

(edge)



Our space usage is $O(n^2)$ for keeping a matrix $A = a(i,j)$.

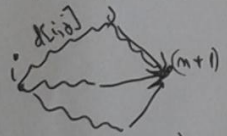
We also define $a(i,j) = \begin{cases} 0, & i=j \\ \infty, & \text{if } v_j \text{ is not reachable from } v_i \\ -\infty, & \text{if there is no } SP(v_i, v_j) \\ k & \text{is the first index on } SP(v_i, v_j). \end{cases}$

We can recover the path by following the entries in the matrix, so the time needed is proportional to the length of the path.

Problem 2.

a). When we add the new node $n+1$, we need to compute the shortest path to and from it for every other node:

$d(i, n+1) = \min \{ d(i,j) + w(j, n+1) \mid 1 \le j \le n \} \quad \forall i : 1 \le i \le n$.

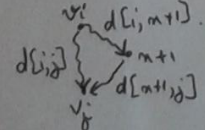


This is the min. of the possible ways to reach $(n+1)$.

Similarly $d(n+1, i) = \min \{ w(n+1, j) + d(j, i) \mid 1 \le j \le n \} \quad \forall 1 \le i \le n$.

Now we need to update the values $d(i,j)$, $1 \le i,j \le n$, since by adding the new node we might be creating a shorter path by going through $n+1$:

$d[i,j] = \min \{ d[i,j], d[i,n+1] + d[n+1, j] \}$.



All these updates take $O(n^2)$ time.

Note that the values $d(n+1, i)$, $d(i, n+1)$ don't need to be updated again if we don't have negative cycles.

Exercises for Unit 23.

Problem 2 b).

If we don't have negative cycles, everything works without problems.
We detect new negative cycles involving $n+1$ by looking at the entry $d(n+1, n+1)$:

$d(n+1, n+1) = \min\{ d(n+1, i) + d(i, n+1) \ , \ 1 \le i \le n \} \cup \{0\}$.

If this is negative, it means we have a negative cycle. We set $d(n+1, n+1) = -\infty$ and we propagate it to the other nodes by recomputing $d[i,j]$. Here we do operations with $-\infty$ in the following way:

$d[i,j] = \min\{ d[i, n+1] + d[n+1, n+1] + d[n+1, j] \}$
$\forall i, j \in \{1, \dots, n+1\}$.

$\min(-\infty, x) = -\infty$

$-\infty + x = -\infty$.

$(-\infty) + \infty = \infty$ (if a not was unreachable, it stays unreachable).

c) Let $G = (V, E)$. We start with G_0 empty and Δ_0 empty and at each step we add a new vertex v_i to obtain G_i and we compute the matrix Δ_i from the matrix Δ_{i-1} like in parts a), b).

In order to recover the path, we can also keep a matrix A of shortest paths like in Problem 1.

Each matrix update takes time $O(n^2)$.
We compute the matrix n times, once for each node.

$\Rightarrow$ Total time $O(n^3)$.

d) This is exactly the Floyd-Warshall Algorithm with a slightly different order of operations.