Maths 3468 quizzes

2: Friday 10/2/12

Note about Floyd-Warhsall: diagonal entries $W[i][i]$ should be initialised to 0.

(1) The runtime of Dijkstra's algorithm was not discussed, but $O(n^2)$ was mentioned in the notes, with a simple implementation, with $O(m + n \log n)$ for more complex implementations.

Simple implementation: T stored as a linked list, for example.

What part of the algorithm produces this non-linear cost?

Answer. It is in selecting a temporary node u, i.e., $u \in T$, with $w(u)$ minimal. With a simple implementation, linear search will be needed with cost $O(|T|)$, and overall cost $O(n^2)$.

(2) It's interesting to modify Dijkstra's algorithm so that shortest paths from s to any v can easily be found. Describe such a scheme.

Answer. Add a parent field to each node. Initially all nodes have null parent. When a node u is selected, and whenever

$$w(u) + w(u, v) < w(v),$$

replace $w(v)$ and let u be the (new) parent of v.

This gives a tree of shortest paths, and the shortest path from s to v can be recovered by tracing back from v to the root.

(3) The same can be done with the Floyd-Warshall algorithm, though not so easily. It is enough to produce a matrix P such that for each i, j, $P[i][j]$ is the *second node* on a shortest weighted path from i to j.

Certainly $P[i][j]$ should be initialised to j or some dummy value, depending on whether or not (i, j) is an edge of the graph.

(i) How should $P[i][j]$ be updated when

$$W[i][j] = W[i][k] + W[k][j]$$

is executed?

Answer.

$$P[i][j] = P[i][k]$$

(ii) How does one construct a shortest path from i to j, using the matrix P? (just an informal description).

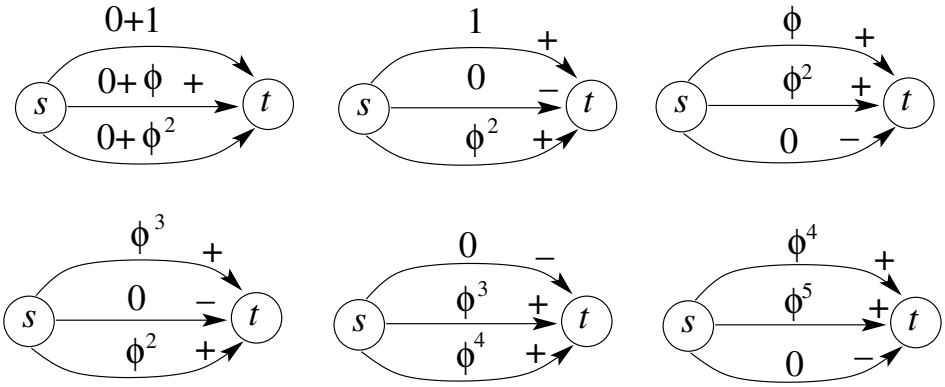
Answer. For a shortest path from i to k, if $i \neq k$, assuming a path exists,

```
j = i;
while ( j != k && j != dummy value )
{
  output j;
  j = P[j][k];
}
```

(Of course if `j` assumes dummy value then there is no such path).

(4) The following is an example of where a flow can be augmented infinitely often, converging to the maximum flow but not reaching it (in finite time). Recall that a flow network can be a multigraph. This makes the construction simpler. The network has edges and capacities as shown, where ϕ is the reciprocal of the golden ratio: $\phi^2 + \phi = 1$. Also, *residual* capacities are shown, i.e., $c(e) - f(e)$ on all edges e — not consistent with the definition for an augmenting path.

Explain what is going on, and how the process can continue indefinitely. (Figure corrected).



Answer. At each step there is one saturated edge, initially the middle one with a flow value of ϕ. Thereafter there is always an augmenting path with two forward edges, residual capacity ϕ^n, ϕ^{n+1}, and one backward edge with flow value $\geq \phi^{n+1}$. The flow can be augmented by ϕ^{n+1} leaving one saturated edge and two others with residual capacities ϕ^{n+1} and ϕ^{n+2}. This is because

$$\phi^n - \phi^{n+1} = \phi^{n+2}$$

Hence there can be infinitely many augmentations.