Exercises, chapters 9–10, solutions

---

1. (a) The following derivation is from p. 224 of the textbook:

$$G = \alpha S + (1-\alpha)\frac{1}{n}\mathbf{e}\mathbf{e}^T$$

so we have:

$$\pi_{k+1}^T = \pi_k^T\left(\alpha S + (1-\alpha)\frac{1}{n}\mathbf{e}_{n\times n}\right)$$

Recall also that:

$$S = H + A = H + \frac{1}{n}\mathbf{w}\mathbf{e}^T$$

so we get:

$$\begin{aligned}
\pi_{k+1}^T &= \pi_k^T\left(\alpha H + \alpha\frac{1}{n}\mathbf{w}\mathbf{e}^T + (1-\alpha)\frac{1}{n}\mathbf{e}\mathbf{e}^T\right)\\
&= \alpha\pi_k^T H + \pi_k^T\left(\alpha\mathbf{w}\mathbf{e}^T\frac{1}{n} + (1-\alpha)\mathbf{e}\mathbf{e}^T\frac{1}{n}\right)\\
&= \alpha\pi_k^T H + \pi_k^T\left(\alpha w + (1-\alpha)\mathbf{e}\right)\mathbf{e}^T\frac{1}{n}\\
&= \alpha\pi_k^T H + \left(\pi_k^T\alpha w + (1-\alpha)\pi_k^T\mathbf{e}\right)\mathbf{e}^T\frac{1}{n}\\
&= \alpha\pi_k^T H + \left(\pi_k^T\alpha w + (1-\alpha)\right)\mathbf{e}T\frac{1}{n}\\
&= \alpha\pi_k^T H + \pi_k^T\alpha\mathbf{w}\mathbf{e}^T\frac{1}{n} + (1-\alpha)\mathbf{e}^T\frac{1}{n}
\end{aligned}$$

(b) The matrix G is dense and very large, so storing G explicitly would require a lot of memory and multiplying the vector π_k^T by G would take a long time. In contrast, the matrix H used in the formula in part (a) is extremely sparse. Consequently, H can be stored more compactly than G, and computing $\alpha\,\pi_k^T\,H$ is much faster than computing $\pi_k^T\,G$. The remaining two terms in the formula can also be computed quickly because the values of $\alpha\,w\,\mathbf{e}^T\,\frac{1}{n}$ and $(1-\alpha)\,\mathbf{e}^T\,\frac{1}{n}$ do not change in successive iterations, so they only need to be calculated once at the beginning of the execution and then retrieved in each iteration.

2. (a) The answer is yes.

Explanation:

The Borda counts are as follows.

Candidate A: $2+2+0+2+1=7$
Candidate B: $1+0+2+1+2=6$
Candidate C: $0+1+1+0+0=2$

By definition, the Borda count winner is candidate A.

Next, since A is preferred to B by most voters (here, three voters vs. two) and A is also preferred to C by most voters (four vs. one), the Condorcet winner is candidate A.

(b) The answer is no.

Counterexample:

Suppose there are three candidates A, B, C and three voters whose preference lists are:
$[A, B, C]$
$[A, B, C]$
$[B, C, A]$
The winner of the proposed voting system is B because B never came in last place, but both A and C did. However, A is preferred to B by most voters and A is also preferred to C by most voters, so A is the Condorcet winner. Thus, the Condorcet criterion is violated.

3. **Algorithm for the Transitive Closure Problem:**

(i) Assign the weight 1 to every edge in E.

(ii) Run the Floyd-Warshall algorithm on the resulting edge-weighted graph to compute *dist*.

(iii) Let G^* be a graph with vertex set V and initially no edges.
For each $x, y \in V$ with $x \neq y$, if $dist[x, y] < |V|$ then insert the directed edge (x, y) into G^*.

(iv) Return G^*.

Explanation: There is a directed path from a vertex x to another vertex y in G (and hence an edge from x to y in the transtive closure of G) if and only if the distance from x to y is less than $|V|$. To check the latter condition, the algorithm above uses the Floyd-Warshall algorithm to obtain the distances between all pairs of vertices in G.

Time complexity analysis: Running the Floyd-Warshall algorithm takes $O(|V|^3)$ time, and the other operations take $O(|E|) + O(|V|^2)$ time. The total time complexity is $O(|V|^3 + |E| + |V|^2) = O(|V|^3)$.

Remark: Faster solutions are also possible by using other techniques such as matrix multiplication or depth-first search.