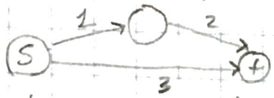
Algorithms Illuminated Part 2: Ch. 9 Andrew Doss

9.1

a) False. Counter example:



b) This reduces down to the question - can two or more distinct powers of 2 sum to another power of 2? put another way, for arbitrary 2^n, an $\exists S \subset \{0,1,2,\dots,n-1\}$ s.t. $\sum_{i \in S} 2^i = 2^n$?

We can show this impossible using induction.

Base case: $n=2$

$$2^0 + 2^1 = 2 < 2^2 = 4$$

For arbitrary n, if $\sum_{i=0}^{n-1} 2^i < 2^n$, then for $n+1$

$$\sum_{i=0}^{n-1} 2^i + 2^n < 2^{n+1} = 2^n + 2^n \checkmark$$

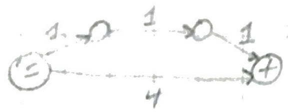
True

c) False, same argument as part a)

d) False, b is correct

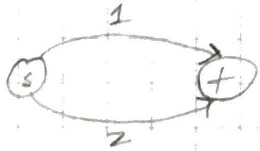
0.3

a) False. Here is a counter-example:



original P is length 3
If we add 10 to each edge, we get that same path having length 33 while the lower path has length 14.

b) False. Here is a counter-example:



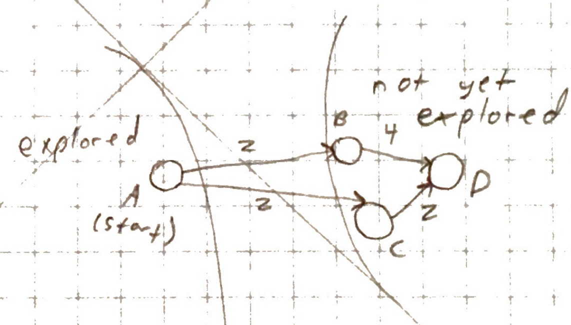
c) True, see arguments for a and b above.

d) True. P receives the minimum distance increase for any path (+10).

9.7 Modify Dijkstra to find the shortest bottleneck of any s-v path for all vertices v and with $O(mn)$ complexity.

Use the straightforward $O(mn)$ implementation of Dijkstra's algorithm, except add the vertex with shortest crossing edge length for each of the n iterations. If the vertex added has the shortest crossing edge length, any other path to the vertex must use another crossing edge with at least as long of an edge length, and therefore no other path can have a shorter minimum bottleneck to that vertex. By applying this repeatedly, it is also ensured that all paths up to the crossing edges have the minimum bottleneck to that vertex.

~~The answer above fails because of ties. If we only look at edge crossing, tie breaking is arbitrary and may later lead to a sub-optimal bottleneck. example:~~



If we pick

The original above is close, but we need to book-keep with $max(len(v_1), weight\ v_1 - v_2)$ to be able to return the minimum bottleneck. The paths from above are correct, however.