1. $max \sum_{S \in \mathcal{S}} y_S$
   subject to,
   $\forall e \in E$, $\sum_{S : e \in \delta(S)} y_S \leq w(e)$
   $\forall s \in \mathcal{S}$, $y_S \geq 0$.

2. It can only increase once, since there must be some e' that $\sum_{S \in \mathcal{S}: e \in \delta(S)} y_S = w(e')$, and you cannot increase it any more.

3. Each dual variable can be increased at most once, and the number of the dual variables is limited, so it will terminate, and each time C will include more nodes until there is a path from s to t, and then it must be a solution to the problem.

4. $val(y*) \leq P*$.

5. It's feasible in the beginning, and as the algorithm going, it will never violate one constraint, so it's still a feasible solution at the end.

6. $val(y) \leq P*$.

7. $w(e) = \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S$.

8. $\sum_{e \in P} w(e) = \sum_{e \in P} \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S$.

9. $\sum_{e \in P} \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S = \sum_{S \in \mathcal{S}} y_S |P \cap \delta(S)|$.

10. We always add y_S, where S is a set containing the starting point s, and P is a single path from s to t, for any set S which contains s, its boundary $\delta(S)$ will only has one intersection with P.

11.

$$\begin{aligned}
\sum_{e \in P} w(e) &= \sum_{e \in P} \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S && (1) \\
&= \sum_{S \in \mathcal{S}} y_S |P \cap \delta(S)| && (2) \\
&= \sum_{S \in \mathcal{S}} y_S && (3) \\
&\leq P* && (4)
\end{aligned}$$

So the output is exactly the optimal solution.

12. Used in Question 10, after pruning, P is a single path from s to t, otherwise, it's a tree with lots of unnecessary edges included.

13. The edge (s, i) should have the smallest weight, and therefore, should be added first into D based on implementation of Dijkstra.

14. vertex j.

15. (j, i), where $i \notin S'$.

16. $l(k) = min(l(k), l(j) + w(j, k))$

17. They are the same, since the update of $l(i)$ and $d(i)$ are the same, and how they include a new vertex are the same.

18. $O(nlogn + m)$, where n is the number of nodes, and m is the number of edges.