PS 6

6-1.

(a)

DAG Relaxation:

topological order: $(c, f, e, b, d, a) \Rightarrow (a, d, b, e, f, c)$

DAG Relaxation order:

① (a, b): $d(a, b) = 2 < \infty$

② (a, d): $d(a, d) = 6 < \infty$

③ (d, b): $d(a, b) = 2 < d(a, d) + w(d, b) = 7$

④ (d, e): $d(a, e) = 6 < \infty$

⑤ (b, c): $d(a, c) = 3 < \infty$

⑥ (b, f): $d(a, f) = 4 < \infty$

⑦ (b, e): $d(a, e) = 5 < d(a, d) + w(d, e) = 6$

⑧ (e, f): $d(a, f) = 4 < d(a, e) + w(e, f) = 7$

⑨ (f, c): $d(a, c) = 3 < d(a, f) + w(f, c) = 5$

Dijkstra:

Dijkstra Relaxation Order:

$d(a, a) = 0$

① (a, b): $d(a, b) = 2$ $\quad d(a, d) = 6$

② (b, c): $d(a, d) = 6$ $\quad d(a, e) = 5$ $\quad d(a, f) = 4$

③ (b, f): $d(a, d) = 6$ $\quad d(a, e) = 5$ $\quad d(a, f) = 4$

④ (b, e): $d(a, d) = 6$ $\quad d(a, e) = 5$

⑤ (a, d): $d(a, d) = 6$

(b)

DAG Relaxation: a: 0 b: 2 c: 3 d: 6 e: 5 f: 4

Dijkstra: a = 0 b: 2 c: 3 d: 6 e: 5 f: 4

6-2.

6-3.

We can model a directed weight graph $G=(C,N)$ where N contains every slope $(c_i, c_j, l_{ij}) \in N$. $|C| = O(n)$, $|N| = O(n)$.

Lemma 1. Without detonating, the graph $G=(C,N)$ forms a dag.

pf. By contradiction. Assume that G has a cycle $C=(v_0, \dots, v_k)$.

Let E denote a relation $E: C \to \mathbb{Z}^+$ such that $E(c)$ for every $c \in C$ represents elevation of c. Then,

$E(v_0) > E(v_1) > \dots > E(v_{k-1}) > E(v_k) = E(v_0)$. This is a contradiction, so there does not exist any cycle in G. □

Corollary 1. A graph G' which is produced by detonating is also a dag.

Thus, we can apply DAG Relaxation.

Find shortest paths using DAG Relaxation starting at L in G and at D in G'

The minimum distance from S to L is

$\min\{\delta(L,S),\ \delta(L,D)+\delta'(D,S)\}$ where $\delta(s,v)$ denotes a shortest path from s to v in G and $\delta'(s,v)$ denotes a shortest path from s to v in G'.

Running Time: $O(|C|+|N|)\cdot 2 = O(|C|+|N|) = O(n+n) = O(n)$.

6-4.

If there exists a negative weight cycle, the total sum of work would be negative infinity. So every cycle is not negative weight cycle. If there exists a positive weight cycle, the total sum of work would be positive infinity. Thus, A force field is conservative IFF for every cycle c, c is zero-weight cycle.
By full-dfs, we can find every cycles exist. There would be O(n) cycles with O(n) particles. So checking each cycle whether is zero takes O(n)-time. We should do this for O(n) cycles. Running time is, thus, O(n) * O(n) = O(n^2)-time.

6-5

Let M denote a graph M = (X, R). X:: = { x | x = x_i}. R ::= { r | r = r_j }.
Let "B : X -> Boolean" that for every x $\in$ X, B(x) = true if there is a gas station at x, otherwise false.
Let "E : X -> Integer" that for every $x_i \in$ X, L(x_i) = e_i.
Let "T : R -> Integer" that for every $r_i \in$ R, T(r_j) = t_i.
$r_i = (u_i, v_i)$.
$\forall r \in R,\ T(r) \geq 0$.
Let G denote G ::= { x $\in$ X | B(x) = true }. |G| = O(n).
|X| = n, |R|= O(n).

(1) Make a directed weighted graph M' = (X, R, w') such taht

$$w'(r \in R) = \begin{cases} T(v) - T(u) & \text{if for some } v, u \in X \wedge (u, v) = r,\ E(v) > E(u) \\ 0, & \text{if } E(v) \leq E(u). \end{cases}$$

(2) Given g, $\forall x_i \in G \cup \{s\}$, get a subgraph M_i of M such taht for a subgraph $M_i = (X_i, R_i)$, X_i ::= { x$\in$X | x is a vertex reachable from x_i in M' such that the sum of weights of its path from x_i to x does not exceed (g - 1) }. R_i ::= { r$\in$R | r is an edge traversed by Diff DFS in M' starting at x_i which the sum of weights of a path does not exceed (g - 1) }.

We can get M_i by Diff DFS, similar to the original DFS but slightly different such that it allows visiting a vertex more than once.

```
  Diff DFS(Adj, w', M , s, g)
  input: Adj is an adjancecy set corresponding to M'. w'∈G'. S is a source, g is an amount
  of gas available to use.
  output: Mi = (Xi, Ri)
  if Mi != ∅
     initialize Gi as an empty set.
  for v∈ Adj(s) - { s }
     if w'(s, v) < g
        put v to Xi ∈ Mi and r = (s, v) to Ri ∈ Mi. // Xi and Ri are sets so that they don't
                                                       allow duplicate of value.
        Diff DFS(Adj, w', Mi, v, g - w'(s, v))
  return Gi
```

Set = { M | M = Diff DFS(Adj of M', w', x_i, g) for every $x_i \in$ G } U { M | M = Diff DFS(Adj of M', w', s, g) }.
Let M_s denote a graph we get by Diff DFS(Adj, w', s, g) and
M_i denote a graph we get by Diff DFS(Adj, w', x_i, g) for every $x_i \in$ G.

③ Let Q denote a relation $Q : X_i \to X$ where $Q(x)$ ::= "Returns $x' \in X$ that x corresponds to x' in original graph M".
For every $m \in$ MSet, $T_i(r)$ ::= "Returns $T(Q(u), Q(v))$ where $r = (u, v)$, $u, v \in m$."

④ For every $m_i, m_j \in$ MSet $\wedge$ $m_i \neq m_j$, add edges from every $u \in m_i - \{x_i\}$ to $v \in m_j$ where $Q(u) = Q(v) \in G$.
By this procedure, we get a connected directed graph $M_F = (X_F, R_F, T_F)$.
$T_F : R_F \to$ Integer, $T_F(r)$ ::= "Returns $T_i(r)$ where $r \in R_i$".

⑤ Make a graph M'_F which is a duplicate of M_F, $M'_F = (V_F, G_F, T'_F)$.

$$T'_F(u, v) = \begin{cases} T_F(u,v) + L(v) - L(u) & \text{if } L(u) < L(v) \\ T_F(u,v) & \text{otherwise.} \end{cases}$$

⑥ Compute SSSPs from s which was a source in Diff DFS to make M_i. Do this for every M MSet - $\{M\}$, using Dijkstra Algorithm in T'_F

⑦ For every shortest path from s to v where $B(Q(v))$ = true, $s = x_i$ for each M_i, $v_i \in X_i$, let π denote such a shortest path $(s = v_1, v_2, \ldots, v_k = v_i)$.
For every $(x_i, x_j) \in R_F$, $x_i \in X_i$, $x_j \in X_j$, $Q(x_i) = Q(x_j)$,
set $T_F(x_i, x_j) = \sum_{m=1}^{k-1} [T'_F(v_i, v_{i+1}) - T_F(v_i, v_{i+1})] \cdot t_G$, where $v_1, \ldots v_k \in \pi$

⑧ It is almost done.
For every vertex $x_s \in X_s$, set $T_F(x_s, x_i) = 0$ for every $x_i \in X_i$, $Q(x_s) = Q(x_i) \in G$.
Make a vertex t' such that it represents the destination.
For every $t_i \in X_i \in$ MSet where $Q(t_i) = t$, add an edge (t_i, t').
Set $T_F(t_s, t') = 0$ where $t_s \in X_s$ and $Q(t_s) = t$.
Let π denote a shortest path from s to t where $s, t \in X_i$ for every $X_i \in$ MSet, $Q(s) = s$, $Q(t) = t$, $\pi = (s = v_1, v_2, \ldots, v_k = t)$.
For every $(t, t') \in R_F$, $t \in X_i \in$ MSet, $Q(t) = t$,
set $T_F(t, t') = \sum_{m=1}^{k-1} [T'_F(v_i, v_{i+1}) - T_F(v_i, v_{i+1})] \cdot t_G$ where $v_1, \ldots, v_k \in \pi$
Then we can do Dijkstra algorithm to find $\delta(s, t')$ where $s \in X_s$, $Q(s) = s$.

Running Time Analysis:

①: $O(|X| + |R|) = O(n)$

②: $O(|R| \cdot |X|) = O(n^2)$

④: $O(|R||X|) = O(n^2)$

⑤: $O(|X|^2 + |R|^2) = O(n^2)$

⑥: $O(|X|(|X| \log(|X|) + |R|)) = O(n^2 \log(n))$

⑦ $O(|X|^2) = O(n^2)$

⑧: $O(|X_F| \log(|X_F|) + R_F) = O(|X|^2 \log(|X|^2) + |R|^2)$
$= O(n^2 \log(n^2) + n^2) = O(n^2 \log(n))$

The case of summation, $O(n^2 \log(n))$.