

Solution ROBOT

Nhận xét đầu tiên là ta sẽ không bao giờ đi lên hay xuống trên 1 cột khi không cần thiết. Do nhảy sang cột khác nếu có thể, hạ cùng lượng độ cao, mất cùng lượng chi phí, lại được chuyển cột khác.

Khi nào là cần thiết. Ta sẽ có 3 cách nhảy.

Từ cột u tại chiều cao H_u , giả sử nhảy bằng 1 cạnh sang cột v tới chiều cao H_v

- Nếu $H_v < 0$ thì ta sẽ đi lên ở cột u 1 đoạn rồi mới nhảy sang cột v ở chiều cao 0, chi phí là $H_u - 0 +$ lượng đi lên.

- Nếu $H_v > \text{Max}H_v$ ta sẽ đi xuống ở cột u 1 đoạn rồi mới nhảy sang cột v ở chiều cao $\text{max}H_v$, chi phí là $H_u - \text{max}H_v$

- Nếu $0 \leq H_v \leq \text{Max}H_v$, nhảy trực tiếp sang, chi phí là $H_u - H_v$.

Từ đây có 1 số nhận xét :

- Nếu dùng chiến thuật trên, khi xuất phát từ u tại H_u , đến được v tại $H_v > 0$ thì chi phí là $H_u - H_v$.

- Nếu đến được v tại chiều cao 0 thì chi phí còn lớn hơn nữa.

- Khi đã ở 0 thì đi tiếp vẫn luôn luôn ở chiều cao 0

- Vậy nên đến được v tại vị trí càng cao thì chi phí càng nhỏ.

Rút ra thuật toán là Dijkstra, $d[u]$ là chi phí ít nhất tới được đỉnh u , tại chiều cao max nhất có thể. (do càng cao chi phí càng ít).

Luôn giữ lại chiều cao lớn nhất tại cùng 1 cột.

Độ phức tạp : $O(M * \log N)$.