Algorithm H
Longest Shortest Paths
Algorithm 해설

최장최단경로들 가운데는 항상 수직 선분 S 혹은 T의 특정한 점들을 연결하는 경로가 있다. 이러한 점들은 S와 T의 끝점이거나, S와 T 위의 점들 가운데 조금씩만 움직여도 최단경로의 길이가 더 커지는 점들이다. 그러한 점들을 찾아내기 위해 아래와 같이 문제를 해결한다.

문제 해결방식

- Plane Sweep 알고리즘을 사용하여 특정한 점들을 아래와 같이 계산한다.
- Sweep line L에는 S로부터 L위의 점까지 최장최단경로를 나타내는 함수를 interval들로 나누어 BBST를 유지한다. 즉, BBST의 각 노드는 sweep line의 점들 가운데 S로부터의 최단경로의 구조가 같은 점들이 하나의 interval이 된다. 최단경로의 구조가 같다는 것은 최단경로의 마지막 선분을 제외한 나머지가 동일하다는 뜻이다.
- Sweep line L이 직사각형의 왼쪽 변 H를 만난 경우를 고려하자.
  - H와 교차하는 interval에 대해서 아래와 같이 경우를 나누어 처리한다.
    - Interval이 H에 포함된 경우, 해당 interval 삭제
    - H가 Interval에 포함된 경우, 위 조건을 만족하도록 해당 interval을 두개의 interval로 나눈다.
    - H와 interval이 일부 교차하는 경우, 위 조건을 만족하도록 해당 interval의 범위를 줄인다.
- Sweep line L이 직사각형의 오른쪽 변 G를 만난 경우를 고려하자.
  - G의 양 끝점에 각각에 대해서 S의 모든 점으로부터의 최단경로를 구간을 나누어 저장한다. $O(n)$ 시간.
  - G의 변에 해당하는 interval 들을 위 조건을 만족하도록 구성하여 BBST에 추가한다. $O(n)$ 시간.
- Sweep line이 T를 만나면 T와 교차하는 BBST의 intervals에 대해 최단경로의 길이를 모두 계산하여 그 가운데 최대값을 결정한다.
- 위의 plane sweep 알고리즘을 S에서 T 방향으로 수평으로 한번, 수직으로 한번 계산하여 최장최단경로를 구한다.
-

이 알고리즘의 시간복잡도는 $O(n^2)$이다.