Divide & Conquer and Search

Thursday, September 21, 2017 4:06 PM

Problem A: 4 thought

- Calculate all 4^3=64 possibilities
- Evaluate string expressions
  - Python

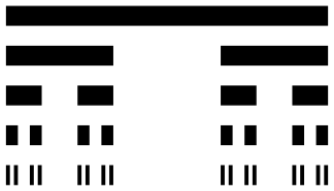
    eval('4 + 4 - 4 / 4') == 7
  - Java

```
ScriptEngineManager mgr = new ScriptEngineManager();
ScriptEngine engine = mgr.getEngineByName("JavaScript");
String result = engine.eval("1+1").toString();
```

- Reference
  - https://en.wikipedia.org/wiki/Eval
  - https://en.wikipedia.org/wiki/Reverse_Polish_notation

Problem B: Cantor

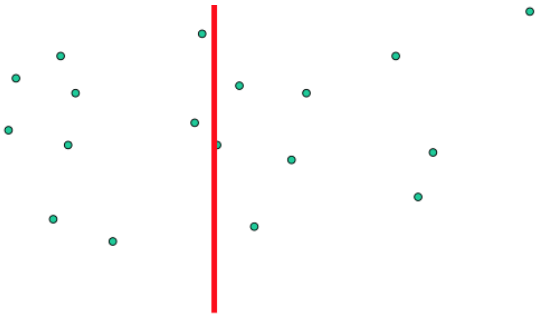
- Cantor set



- Algorithm
  - 0~1/3: ?
  - 1/3~2/3: out of Cantor set
  - 2/3~1: ?
  - Calculate until repeat
- Question
  - Maximum number of digits until repeat???
- Reference
  - https://en.wikipedia.org/wiki/Cantor_set

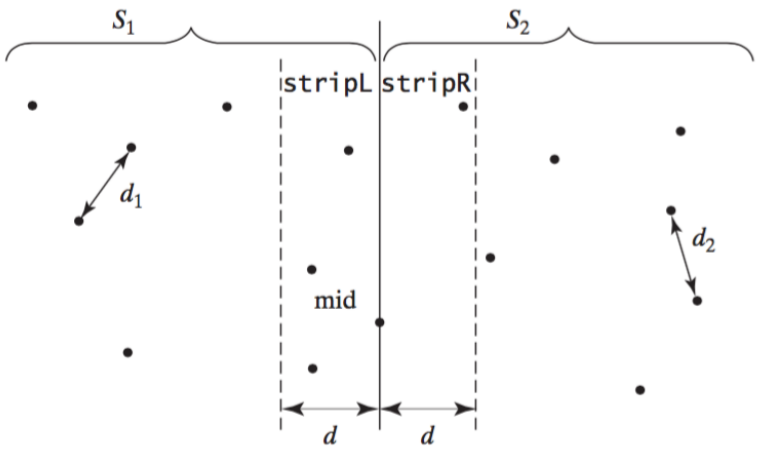
Problem C: Closest Pair

- Algorithm: Binary Search
  1. Divide all points by x-median



Divide by x-median

2. Find the minimum distance of left and right division d_1 *and* d_2
3. Let $d = min(d_1, d_2)$
4. Consider the strip area below



At most 6 points in one's neiborhood, because no two points can be within distance d.

- Reference:
  - http://www.geeksforgeeks.org/closest-pair-of-points/
  - https://en.wikipedia.org/wiki/Closest_pair_of_points_problem
  - http://slideplayer.com/slide/4497080/

Question D: Gregory the Grasshopper

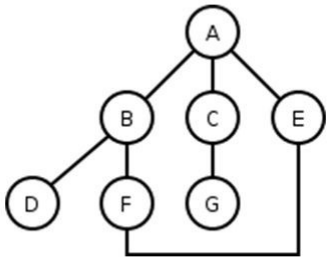
BFS pseudocode

- Pseudo-code for breadth-first search:

```
bfs(v1, v2):
   List := {v1}.
   mark v1 as visited.

   while List not empty:
      v := List.removeFirst().
      if v is v2:
         path is found.

      for each unvisited neighbor v_i of v
      where there is an edge from v to v_i:
         mark v_i as visited.
         List.addLast(v_i).
```



path is not found.

- Reference
  - https://en.wikipedia.org/wiki/Breadth-first_search
  - http://slideplayer.com/slide/4318446/

Question E: Medals

- Mathematical Understanding
  - Weight vector: $w = (wieght\,for\,gold, weight\,for\,silver, weight\,for\,bronze)$
  - Medal vector: $m = (gold,\ silver,\ bronze)$
  - Result: $w \cdot m$
  - Total 6+6+1 ways of ranking.
- Example
  - $if\ i = 0,\ k = 1$
  - $\frac{1}{n_0} = [0, 1, \dots, n] \rightarrow [0, 1, \dots, n]$
  - $\frac{1}{n_1} = [0, 1, \dots, n] \rightarrow \left[0, \frac{1}{n}, \frac{2}{n}, \dots, 1\right]$
- Note
  - If there is no Canada, it cannot win! (Be careful about input specs and edge cases!)

```
weight[0][i] = gold + silver + bronze;
weight[1][i] = 10000 * gold + 100 * silver + 1 * bronze;
weight[2][i] = 10000 * gold + 1 * silver + 100 * bronze;
weight[3][i] = 100 * gold + 10000 * silver + 1 * bronze;
weight[4][i] = 100 * gold + 1 * silver + 10000 * bronze;
weight[5][i] = 1 * gold + 100 * silver + 10000 * bronze;
weight[6][i] = 1 * gold + 10000 * silver + 100 * bronze;
```

Also, weights of (100, 1, 1), (1, 100, 1), (1, 1, 100), (100, 100, 1), (100, 1, 100), (1, 100, 100).