CSCI 406: AlgoBOWL

Reminder: You are NOT allowed to consult the internet to solve this problem.

1 Problem Description

You are given n points $p_1, p_2, ..., p_n$ in three-dimensional space. Your task is to partition the n points into k sets so that the maximum distance between any two points in the same set is minimized. Your algorithm should use the Manhattan distance metric, which for two points (x_1, y_1, z_1) and (x_2, y_2, z_2), is defined as $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|$.

The input format includes n, k, and the coordinates of the n points. A very simple example follows. Simple because all of the points are on the x-axis. This will not be true in general. I've included comments for your understanding. Comments should not be included in the actual inputs:

```
5   // n
2   // k
1  0  0  // first point
2  0  0  // second point
3  0  0  // third point
8  0  0  // fourth point
9  0  0  // fifth point
```

Let's assume that the points are named p_1, p_2, p_3, p_4 and p_5. They are to be divided into $k = 2$ sets based on the Manhattan metric. A possible partition is $\{p_1, p_3, p_5\}$: $\{p_2, p_4\}$. The cost of this partition is 8 because the Manhattan distance between p_1 and p_5 is 8 and this is the largest distance between any pair of points in a set. Another possible partition is $\{p_1, p_2, p_3\}$: $\{p_4, p_5\}$. It's cost is 2 which is the Manhattan distance between p_1 and p_3. Clearly this second partition is better than the first.

Input Restrictions:

- $3 \leq n \leq 1000$ (i.e., any input contains at least 3 and at most 1000 points).
- $2 \leq k \leq 20$ (i.e., the number of sets is between 2 and 20).
- The coordinates of all points are integers in $[-1000, 1000]$. (Note that this means that the Manhattan distance between any pair of points will also be an integer.)

Output Format: Line 1 of your output will contain the maximum distance between any two points in a set. Line 2 will contain point IDs assigned to the first set and Line 3 will contain point IDs assigned to the second set. To illustrate, the output for the second split in the example is

```
2
1 2 3
4 5
```

Note: This problem is NP-complete, which means that it is unrealistic to expect that your algorithm will compute an optimal solution in a reasonable time frame. Please keep this in mind as you work on this project.

2 Deliverables

Your group has three tasks:

1. Develop as good an algorithm as you can that accepts a valid input and produces a valid output.
2. Create an input within the parameters specified above that will challenge the other groups.
3. Develop a tool that verifies the other groups' outputs. What does this mean? The purpose of the verifier is to examine the outputs that other groups compute on your input and check that the cost they are reporting is consistent with their solution. (The verifier is not checking whether the solution provided by the other group is optimal.)

3 Grades

Overview: AlgoBOWL is 10% of your total grade. Your grade will be based on the following:

1. Does your software work and produce correct (i.e., valid, not optimal) solutions in a reasonable amount of time? [$\approx 75\%$ of the grade].
2. How good were your inputs at stress-testing the other groups' algorithms? [$\approx 10\%$ of the grade]. An incorrectly formatted input or one that does not adhere to the specs can result in your group forfeiting these points!
3. How do the solutions produced by your algorithm compare to those of other groups? [remaining $\approx 15\%$].
4. Your effort distribution within the group.

4 Logistics

You will use the AlgoBowl tab at `connect.mines.edu` for the three steps below. To do this, you have to register on `connect`. Hopefully you did this already in the first week of class.

1. Step 1: Upload your group's input by 6pm on Oct 4.
2. Step 2: Upload your group's output on *each of the other group's inputs* by 6pm on Oct 5.
3. Step 3: Verify all of the other group's outputs on your input by 6pm on Oct 6.
4. Step 4: Upload your group's effort distribution by 6pm on Oct 6.

Good Luck!!!