**Test analysis**

**Programming exercises – Algo training**

**knapsack**

Tests 1-3

Simple examples

Test 4

Large number of objects (N=19)

Test 5

Let's examine the following greedy approach:

For every object, we compute its **density**, defined as . As long as there is still place in the knapsack, we look at all the elements that have lower

weight than the remainder of the capacity, and choose the one with the

highest density.

This test proves that the greedy method is not optimal.

Test 6

Big capacity (C=100).

**Closest pairs**

Tests 1-5

Input samples are increasingly growing: N = 6, 12, 100, 10000, 250000

**Double bit switch**

Tests 1-4

Input samples are increasingly growing: N = 4, 10, 1000, 100000

**String chain**

Tests 1

Simple chainable input

Test 2

Simple non-chainable input

Test 3

This input is not chainable, even though it's made of 2 chainable couples.

Test 4

Big input (N = 26000) that is chainable

Test 5

Big input (N=26000) that is not chainable. This is the same list of words from test 4, with the word 'perel' added.

Test 6

Hugh input (N=23400, every string length is between 90 to 110)

**Shortest path amount**

Tests 1-2

Simple inputs (V = 3, 7) of directed graphs. Test 2 is the same as shown in the example.

Tests 3-4

Simple inputs (V=20 for both) of undirected graph. Both has the same graph with different sources.

Test 5-6

Big undirected connected graphs with V=1000.

* In test 5 every node degree is 20, total of E=10000
* In test 6 every node degree is 120, total of E=60000

Test 7

Undirected not-connected graph with V=500.

* 400 nodes create a connected component in which every node degree is 8. The source is in this component.
* 100 nodes create a connected component in which every node degree is 35.

Tests 8-10

Huge undirected connected graph with V=10000.

* In test 8 every node degree is 25, total of E=125000
* In test 9 every node degree is 50, total of E=250000
* In test 10 every node degree is 120, total of E=600000

**Rotations**

Tests 1-4

Simple examples with V=3, 6, 11, 11

Test 5

Big tree with V=100001. The tree is built from a root with 100000 children.

Test 6

Big tree with V=100001. The tree is built as a line – the root has only one child, that has only one child… and so on for 100000 edges.

Test 7

Big tree with V=100000 built randomly.

**Dijkstra**

Test 1

Randomly built graph with V=100, E=600. Every edge has a weight attribute between 0-200.

**Reverse edges**

Tests 1-3

Simple graph with V=2, E=1, 1, 0.

* In test 1, the edge is in the desired direction.
* In test 2 the edge is not in the desired direction and must be flipped.
* In test 3, there is no edge between the nodes – path is not possible.

Test 4

Simple graph with V=6, E=7. This is the graph given as example.

Test 5-7

Graph with V=300, E=450, 450, 150.

* In tests 5-6, every node has total degree of 3. The tests differ in the given source and target.
* In test 7, every node has total degree of 1.

Test 8

Simple graph with V=4, E=4. Between nodes 1, 2 there is a bidirectional edge (or two directional edges in both directions). This test exists to make sure that in the case of bidirectional edge, both the edges in the new graph will be given weight 0 and not weight 1.

Test 9

Big graph with V=80000, E=120000. Every node has a total degree of 3.

Test 10

Huge graph with V=300000, E=450000. Every node has a total degree of 3.

**Currency exchange**

Tests 1-6

Input samples are increasingly growing: N = 2, 2, 10, 10, 300, 500

**Disjoint paths**

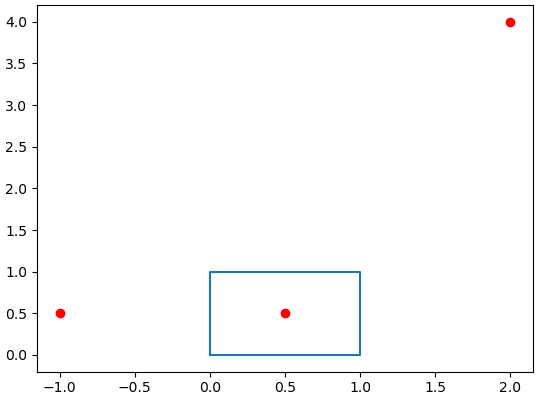
Tests 1-5

Input samples are increasingly growing: V=4, 8, 80, 700, 10000

**Contains**

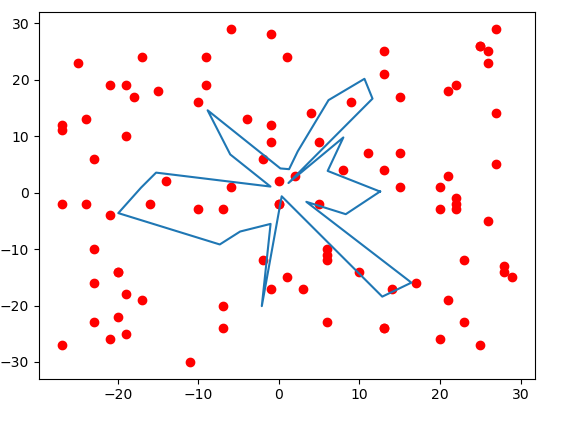
Test 1

Simple polygon



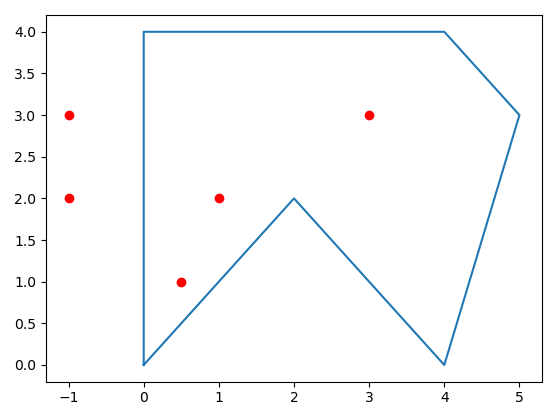
Test 2

Complex polygon



Test 3

Simple polygon. This test check correct dealing with lines passing through nodes.



**Discount**

Tests 1-7

Input samples are increasingly growing: V=4, 8, 200, 5000, 5000, 10000, 10000.

Tests 4-5 has the same graph with different k values, and 6-7 as well.

**Articulation points**

Test 1

Simple graph (V=5, E=6) with single articulation point.

Test 2

Simple graph (V=9, E=10) with 3 articulation points

Test 3

Graph with 23 nodes, made of 2 connected components of 11 and 12 nodes each. There is a single edge connecting node 7 (from the first component) and 23 (from the second) to create a connected graph. Those are the only articulation points.

Tests 4-5

* In test 4 we have a graph with 1201 nodes connected as a row (every node connected to the one before and after it, 0 is connected only to 1 and 1200 only to 1199). Every node is an articulation point except of 0 and 1200.
* In test 5 we have the same graph but now built as a close circle (we add the edge (0, 1200). Now there is no articulation point in the graph.

Tests 6-7

* Graph with 4003 nodes. The graph is composed from 4 components of 1000 nodes each (0-999, 1000-1999, 2000-2999, 3000-3999) in which every node has a degree of 12. In addition, 3 more nodes (4000, 4001, 4002) exist, and they are the 'core' of the graph – the connect the different components. We have 1 node from every component that is connected to either 4000 or 4001 (547 and 1321 to 4000, 2967 and 3433 to 4001) and 4002 is connected to both of 4000, 4001. This way all chosen nodes from the components (547, 1321, 2967, 3433) and core nodes (4000, 4001, 4002) are articulation points.
* A more complex graph with the same idea behind the graph in test 6, this time with 7 connected components and 8 core nodes, total of V=7008, E=42015. This time not all the chosen vertices/core nodes are articulation points.

**Sport elimination**

Tests 1-5

Input samples are increasingly growing: N=3, 6, 30, 40, 50

Test 6

Small input with N=5. In this input every team can still qualify to the playoffs, but teams 0, 1 need a specific set of events for it to happen:

* Currently team 0 has no wins and can achieve at most 2 wins.
* If team 1 wins it only game left, teams 2-3 lose their game left (that means that team 4 wins the matches against 2,3 and loses the matches to 0,1) we will have an overall tie since all the teams will have exactly 2 wins, so every team will qualify including team 0.

The same explanation goes for team 1.

(The rest of the teams can qualify as well, but can achieve more than 2 wins so they don't require the overall tie)

**Road lamp**

Tests 1-4

Small inputs (that both solutions will fit):

* In test 1, N=4, K=3, M=14
* In test 2, N=3, K=12, M=100
* In test 3, N =12, K=6, M=465
* In test 4, N=12, K=24, M=350

Test 5

In this test, the size of the input is N=50000, K=300, M=1253494625.

Any solution with runtime based linearly on N will fail to return an answer within the time limit. Yet, the solution with runtime of KlogN will work perfectly.

Test 6

In this test, the size of the input is N=300, K=500000, M=722443228.

Any solution with runtime based linearly on K will fail to return an answer within the time limit. Yet, the solution with runtime of NlogM will work perfectly (even though M is huge, because its inside the log).

**Poster**

Tests 1-5

Input samples are increasingly growing: N = 3, 4, 7, 50, 5000

Test 6

Input with N=1000 but heights values are big (between 10^6 to 10^9)

Tests 7-10

Input samples are increasingly growing: N = 20000, 75000, 200000, 200000.

In tests 7 and 10, heights values are non-integers.