HeroPath

Specification v1.00

Welcome to Code Weekend #1!

Timeline

June 7th at 21:00 UTC – start of the contest.

June 8th at 21:00 UTC – new version of the statement is released. More tests are added.

June 9th at 21:00 UTC – end of the contest.

Tests

In the first half of the contest, there are 25 tests, which you need to solve. The archive with tests is available in the Discord channel: https://discord.gg/M6pG5zp3DF. Each test is encoded in json format. For each test you need to create answer in json format and submit to the system. You can submit your answers manually through the web interface or via Python API (see https://codeweekend.dev/#/api). You don't need to submit the code, which you used to produce the answer.

Note that you can visualize tests and your solutions on the Dashboard tab (https://codeweekend.dev/#/dashboard).

Intro

There is a world of size $width \times height$, where all objects can only exist at integer coordinates $0 \le x \le width$ and $0 \le y \le height$.

You control a hero in this world which has the following attributes:

- (x_h, y_h) - coordinates
- $base\_speed$ - base movement speed
- $base\_power$ - base attack power
- $base\_range$ - base attack range

In this world, there are also monsters with the following attributes:

- (x_m, y_m) - coordinates
- hp - health points
- $gold$ - the amount of gold the hero will receive if the monster is defeated
- exp - the amount of experience the hero will gain if the monster is defeated

Game

The game consists of num_turns turns. On each turn, the hero can:

- Move to (x_t, y_t). During each turn, the hero can move from (x_h, y_h) to any (x_t, y_t) provided that $(x_h - x_t)^2 + (y_h - y_t)^2 \le (\text{hero speed})^2$ and $0 \le x_t \le width$, $0 \le y_t \le height$. See the 'Experience and levels' section below for the definition of hero speed.
- Attack a monster if the distance between the hero and a monster does not exceed the hero's attack range. The attack reduces the monster's health points by the hero's attack power. See the 'Experience and levels' section below for the definitions of attack range and power. If the monster's health points drop below or equal to 0, it dies, and the hero receives the corresponding amount of gold and experience.

Experience and levels

Gaining experience leads to hero's level increase. The hero begins at level 0. To advance to level 1, the hero needs to earn 1000 experience points. For level 2, an additional 1100 experience points are required. For levels 3, 4, and 5, additional 1300, 1600, and 2000 experience points respectively are required. For level L, an additional $1000 + L \cdot (L - 1) \cdot 50$ experience points are required after reaching the previous level. Levels can be obtained infinitely.

A hero of level L:

- moves with a speed of $\lfloor base\_speed \cdot (1 + L \cdot \frac{level\_speed\_coeff}{100}) \rfloor$,
- attacks with a power of $\lfloor base\_power \cdot (1 + L \cdot \frac{level\_power\_coeff}{100}) \rfloor$,
- has an attack range of $\lfloor base\_range \cdot (1 + L \cdot \frac{level\_range\_coeff}{100}) \rfloor$.

Here $base\_speed$, $base\_power$ and $base\_range$ are hero's attributes. The leveling coefficients for speed, power, and range are provided in the test data.

Scoring

The goal in HeroPath is to accumulate as much gold as possible within the allotted num_turns turns for each test. Your performance in a test is evaluated based on the gold collected, which determines your score in the following manner:

- **Submission Validity**: A submission is considered valid only if the JSON file is correctly formatted and all actions taken by the hero adhere to the game rules. Specifically, a submission will be deemed invalid if:
  - The JSON file is malformed or incorrectly formatted.
  - The hero attempts to move beyond his movement range or outside the boundaries of the map.
  - The hero tries to attack a monster that does not exist, is already dead, or is outside the hero's attack range.
- **Submission Score**: The score for a valid submission is the total amount of gold collected during that attempt. If a submission is invalid, the score for that attempt will be zero.
- **Absolute Score**: For any given test, your absolute score is the highest amount of gold collected across all your valid submissions for that test.
- **Relative Score**: This is calculated as the ratio of your absolute score to the highest absolute score achieved by any participant in the same test. It is expressed as:

$$\text{Relative Score} = \frac{1000 \cdot \text{Your Absolute Score}}{\text{Highest Absolute Score among all participants}}$$

- **Total Score**: Your total score across the competition is calculated by summing your relative scores for all tests.

Input

The test description is contained in a JSON file with the following structure:

```
{
  "num_turns": <number of turns>,
  "width": <world width>,
  "height": <world height>,
  "start_x": <hero starting X coordinate>,
  "start_y": <hero starting Y coordinate>,
  "hero": {
    "base_speed": <base speed of the hero>,
    "base_power": <base power of the hero>,
    "base_range": <base attack range of the hero>,
    "level_speed_coeff": <bonus to hero speed for each level>,
    "level_power_coeff": <bonus to hero power for each level>,
    "level_range_coeff": <bonus to hero attack range for each level>
  },
  "monsters": [
    <list of monsters>
  ]
}
```

Each monster is represented as follows:

```
{
  "x": <monster X>,
```

```
  "y": <monster Y>,
  "hp": <monster hp>,
  "exp": <experience gain for killing this monster>,
  "gold": <gold gain for killing this monster>
}
```

Output

The output for the test should be a JSON in the following format:

```
{
  "moves": [
    move1,
    move2,
    ...,
    moveK
  ]
}
```

Each move can be either of the following types:

```
{
  "type": "move",
  "target_x": <X>,
  "target_y": <Y>
}
```

or

```
{
  "type": "attack",
  "target_id": <0-based monster index from the input>
}
```

For visualization purposes, you can add comments to the moves, which will be shown in the visualizer. Simply include *"comment"* in any move like this:

```
{
  ...
  "comment": "Target: 234, estimated value: 0.5"
}
```

Example

Here you can find an example of an input file along with a possible output file for that input. Note that this test is not present in the testing system. The score for this example output is 664. Please note that after killing monsters 5, 0, and 3, the hero had leveled up.

Sample input

```
{
  "hero": {
    "base_speed": 7,
    "base_power": 50,
    "base_range": 7,
    "level_speed_coeff": 25,
    "level_power_coeff": 25,
    "level_range_coeff": 25
  },
  "start_x": 18,
```

```
  "start_y": 16,
  "width": 50,
  "height": 50,
  "num_turns": 12,
  "monsters": [
    {
      "x": 3,
      "y": 30,
      "hp": 37,
      "gold": 158,
      "exp": 510
    },
    {
      "x": 12,
      "y": 50,
      "hp": 53,
      "gold": 177,
      "exp": 680
    },
    {
      "x": 48,
      "y": 22,
      "hp": 56,
      "gold": 194,
      "exp": 540
    },
    {
      "x": 6,
      "y": 26,
      "hp": 45,
      "gold": 171,
      "exp": 380
    },
    {
      "x": 46,
      "y": 42,
      "hp": 105,
      "gold": 371,
      "exp": 180
    },
    {
      "x": 21,
      "y": 20,
      "hp": 18,
      "gold": 141,
      "exp": 230
    }
  ]
}
```

Sample output

```
{
  "moves": [
    {
      "type": "move",
      "target_x": 15,
```

```
      "target_y": 18
    },
    {
      "type": "attack",
      "target_id": 5
    },
    {
      "type": "move",
      "target_x": 9,
      "target_y": 21
    },
    {
      "type": "move",
      "target_x": 6,
      "target_y": 24
    },
    {
      "type": "attack",
      "target_id": 0
    },
    {
      "type": "attack",
      "target_id": 3
    },
    {
      "type": "move",
      "target_x": 13,
      "target_y": 23
    },
    {
      "type": "move",
      "target_x": 21,
      "target_y": 23
    },
    {
      "type": "move",
      "target_x": 29,
      "target_y": 23
    },
    {
      "type": "move",
      "target_x": 37,
      "target_y": 23
    },
    {
      "type": "move",
      "target_x": 43,
      "target_y": 23
    },
    {
      "type": "attack",
      "target_id": 2
    }
  ]
}
```

Static picture of sample input and output

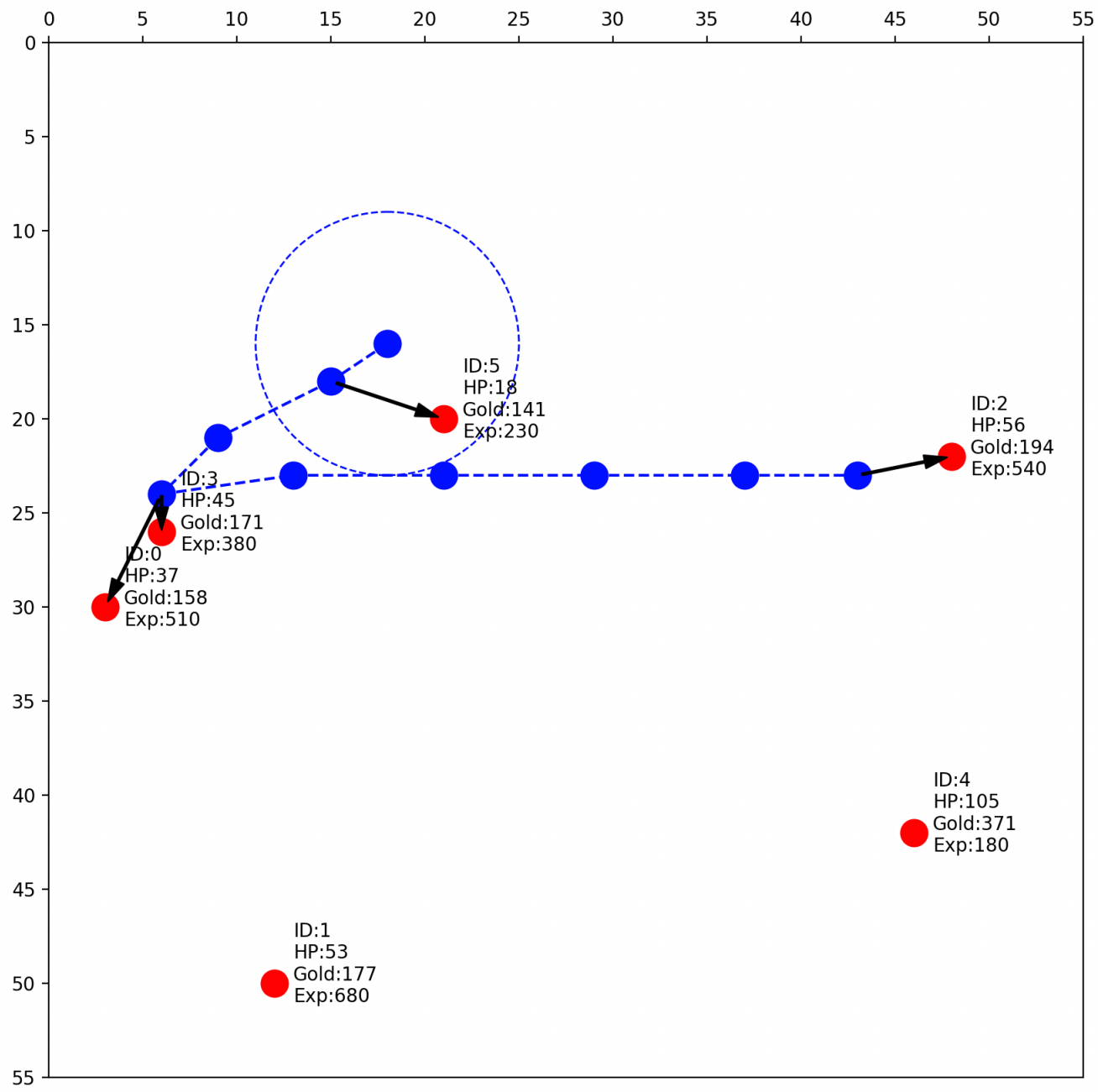

Figure 1: Sample vis