Algorithm used:

1. Vector is sorted in ascending order

2. least and sec\_least, represent the hikers taking the least amount of time to cross the bridge. i.e. element a[0] and a[1] in the sorted array.

3. In every odd numbered iteration, least and sec\_least cross the bridge and least comes back with the torch. Hence time taken to come back is kept minimal. (a[0] + a[1])

4. In every even numbered iteration, last 2 hikers with highest times cross the bridge together. And sec\_least returns back with the torch(a[1]) . Keeping return time minimal.

5. This is continued till only least and sec\_least hikers are remaining and they cross over the bridge at sec\_least's time (a[1])

6. Adding all these times together produces the least amount of time taken by all hikers to cross the bridge.

The solution parses through the sorted array once and calculates the least time required.

Time taken is O(N)

Crossing the bridge algorithm explained with an example:

1. Based on the length of the bridge to be crossed and speed of the hikers, time taken by each hiker is calculated and put into an array.

Example:

Length of bridge = 100ft

Speed of hikers: [100ft/min, 50ft/min, 20ft/min, 10ft/min]

Array a is an array of time taken = {1, 2, 5, 10}

**Consider A:1 B:2 C:5 D:10**

|  |  |
| --- | --- |
| A and B cross the bridge together and A returns | 2 + 1 |
| C and D cross together and B returns | 10 + 2 |
| A and B cross together | 2 |

Total time taken : 17 mins

1. Length of bridge = 250ft

Speed of hikers: [100ft/min, 50ft/min, 20ft/min, 10ft/min, 2.5ft/min]

Array a is an array of time taken = {2.5, 5, 12.5, 25, 100}

**Consider A: 2.5 B:5 C:12.5 D:25 E:100**

|  |  |
| --- | --- |
| A and B cross together and A returns | 5 + 2.5 |
| D and E cross together and B returns | 100 + 5 |
| A and C cross together and A returns | 12.5 + 2.5 |
| A and B cross together | 5 |

Total time taken : 132.5 mins.

The input YAML file : input.yml.

The YAML structure represents, each bridge and the speed of hikers crossing that bridge.

At each new bridge, new hikers join the hike.

---

bridges:

100: (Bridge1)

- 100

- 50 Hikers crossing bridge1

- 20

- 10

250: (Bridge 2)

- 2.5 Hikers joining at bridge 2

150:

- 25

- 15

100:

- 75

100:

- 75