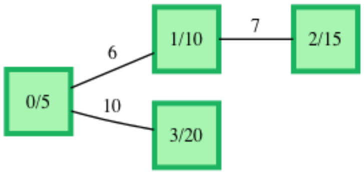
Fidelity International Python Developer Hiring Test - Exp... 16m to test end 0/4 Attempted Vivek Ratnaparkhi

1
2
3
4

City Attractions

You have just arrived in a new city and would like to see its sights. Each sight is located in a square and you have assigned each a *beauty* value. Each road to a square takes an amount of time to travel, and you have limited time for sightseeing. Determine the maximum value of *beauty* that you can visit during your time in the city. Start and finish at your hotel, the location of sight zero.

For example, there are $n = 4$ squares with sights in the city, possibly including something of beauty at your hotel. There are $m = 3$ bidirectional roads that connect them, that join squares $u = [0, 1, 0]$ with squares $v = [1, 2, 3]$. Squares are numbered 0 to $n\text{-}1 = [0 - 3]$. Times to travel each road are $t = [6, 7, 10]$. The beauty values for each square are $beauty = [5, 10, 15, 20]$, and the time you have for sightseeing is $max\_t = 30$. Arrays u, v and t are aligned by index and it takes no time to visit a sight. The *beauty* array indices line up with their square numbers. A graphical representation follows:



Each square is labeled as *[square number]/[beauty value]*. Always start and end at square *0*, the hotel. If you take the top path, from square, $0 \rightarrow 1 \rightarrow 2 \rightarrow 1 \rightarrow 0$ it takes $6 + 7 + 7 + 6 = 26$ minutes and the beauty sum is $5 + 10 + 15 = 30$. If you take the bottom path, it takes $10 + 10 = 20$ for a roundtrip, and the beauty sum is $5 + 20 = 25$. There is not enough time to do both paths, so the best you can do is take the top path and see *30* beauty units worth of sights. Note that you only count the beauty value of a sight on your first visit.

Function Description

Complete the function *findBestPath* in the editor below. The function must return an integer, the maximum sum of beauty values of squares you will visit.

findBestPath has the following parameter(s):

n: an integer, the number of sights in the city

m: an integer, the number of connecting roads

max_t: an integer, the amount of time for sightseeing

beauty[beauty[0]...beauty[n-1]]: integer array, the beauty values you have assigned to each sight

u[u[0]...u[m-1]]: integer array, the starting sight location for each bidirectional road

v[v[0]...v[m-1]]: integer array, the ending sight location for each bidirectional road

t[t[0]...t[m-1]]: integer array, the travel time for each bidirectional road

Constraints

- $1 \le n \le 1000$
- $1 \le m \le 2000$
- $10 \le max\_t \le 100$
- $0 \le u[i], v[i] \le n - 1$
- $u[i] \ne v[i]$
- $10 \le t[i] \le 100$
- $0 \le beauty[i] \le 10^8$
- No more than *4* roads connect a single square with others.
- Two squares can be connected by at most one road.

Input Format For Custom Testing

Sample Case 0

Sample Input 0

1
2
3
4

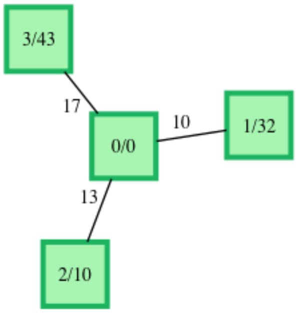
```
4
0
32
10
43
3
0
2
0
3
1
0
3
3
10
13
17
```

Sample Output 0

```
43
```

Explanation 0

```
n = 4
m = 3
max_t = 49
beauty = {0, 32, 10, 43}
u = {0, 2, 0}
v = {1, 0, 3}
t = {10, 13, 17}
```



You can visit squares in the following order: *0*, *3*, *0*. It will take *17 + 17 = 34* minutes. The sum of the beauties of the visited squares will be *0 + 43 = 43*. You could visit both of the other sites in the time given, *10 + 10 + 13 + 13 = 46*, but you would only see sights that total *32 + 10 = 42* beauty value.

Sample Case 1

Sample Case 2

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour ✕

For help on how to read input and write output in Python 3, click here. ✕

Original Code Python 3

```
#!/bin/python3 ···

#
# Complete the 'findBestPath' function below.
#
# The function is expected to return an INTEGER.
# The function accepts following parameters:
#  1. INTEGER n
#  2. INTEGER m
#  3. INTEGER max_t
```

Fidelity International Python Developer Hiring Test - Exp...

16m
to test end

0/4 Attempted

Vivek Ratnaparkhi

1

2

3

4

```
#

def findBestPath(n, m, max_t, beauty, u, v, t):
    # Write your code here

if __name__ == '__main__':⋯
```

Line: 10 Col: 1

□ Test against custom input

Run Code

Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

About Privacy Policy Terms of Service