

ADS HW4 Report

404261476 資工四甲 楊培澤

Problem : Dynamic Frog

Solution :

I use dynamic programming to solve this problem. It is quite same as problem two. The distance is equal to problem 2's cost.

Discussion :

Probably Greedy method is a better choice.

Code :

```
if __name__ == '__main__':
    # dir
    input_dir = './data/Dynamic Frog.in'
    output_dir = './data/Dynamic Frog.out'

    # load all data
    tmp = [line.strip('\r').strip('\n').split(' ') for line in open(input_dir, 'r').readlines()]
    # tmp = [
    #     ['3'],
    #     ['1', '10'],
    #     ['B-5'],
    #     ['1', '10'],
    #     ['S-5'],
    #     ['2', '10'],
    #     ['B-3', 'S-6']
    # ]
    case = int(tmp[0][0])
    ans = []

    tmp_count = 1
    for c in range(case):
        n, d = int(tmp[tmp_count][0]), int(tmp[tmp_count][1])
        tmp_count += 1
        arr = [1e9 for i in range(310)]
        result = [[1e9 for i in range(310)] for j in range(310)]

        step_count = 0
        for i in range(n):
            action = tmp[tmp_count][i].split('-')
            arr[step_count] = int(action[1])
            step_count += 1
            if action[0] == 'B':
                arr[step_count] = int(action[1])
                step_count += 1
            tmp_count += 1
```

```

arr[step_count+1] = 0
arr[step_count+2] = d
arr[step_count+3] = d

sc = step_count+3
arr.sort()
# for i in range(sc):
#     print(' %d' % (arr[i]))

result[0][0] = 0
for i in range(sc):
    for j in range(sc):
        idx = max(i, j) + 1
        result[i][idx] = min(result[i][idx], max(result[i][j], arr[idx]-arr[j]))
        result[idx][j] = min(result[idx][j], max(result[i][j], arr[idx]-arr[i]))

# print('Case %d: %d' % (c+1, result[sc-1][sc-2]))
ans.append(result[sc-1][sc-2])

# check accuracy
result = ans
error = []
ground_truth = open(output_dir, 'r').readlines()
if len(result) != len(ground_truth):
    print('Result count inconsist : result length = %d, ground_truth length = %d'
          % (len(result), len(ground_truth)))
else:
    count = 0
    for i in range(len(result)):
        word_list = ground_truth[i].strip('\r').strip('\n').split(' ')
        if result[i] == int(word_list[2]):
            count += 1
        else:
            error.append({
                'line': i+1,
                'result': result[i],
                'ground_truth': int(word_list[2]),
            })
    if count == len(result):
        print('Match output file')
    else:
        print('Some error in result')
        print(error)

# make output file
filename = ''.join(['./', 'output.txt'])
with open(filename, 'w') as f:
    for i in range(len(result)-1):
        f.write('Case %d: %d\n' % (i+1, result[i]))
    f.write('Case %d: %d' % (len(result), result[len(result)-1]))
print('Output predict file...')

```