

# 实验9 贪心选择补充水量问题

---

姓名：邵宁录

学号：2018202195

## 目录

---

1. [问题描述](#)
2. [算法基本思路](#)
3. [算法复杂度分析](#)
4. [源码](#)
5. [运行结果截图](#)
6. [问题与总结](#)

## 一、问题描述

---

在滑完全程的前提下求最小的补水次数。

输入：补水点位置的数组  $stops$  和 可以装的水上限  $m$ ；

输出：最少的补水次数和补水位置；

## 二、算法基本思路

---

该算法的基本思路较为简单，但证明过程稍微麻烦一点。

思路就是非常普通的贪心思路，只需在滑行到最远距离且刚好不会消耗完水的那个补水点补水，就能够解得最少的补水次数。

该问题存在最优子结构的证明如下：

显然，该问题存在最优子结构。因为一旦我们选择了第一个补水点  $p$ ，那么我们接下来需要解决的子问题就变成了要从  $p$  开始的一个问题。

用剪切-粘贴法可以很快的证明这是正确的：假设  $p$  是最优选择，否则存在一个更优的  $p'$ ，使得整个问题需要停下的补水点更少。这与  $p$  是最优的矛盾，因此可以剪切-粘贴替换掉  $p$ 。得证。

贪心算法正确性证明如下：

不妨设  $O$  是任意一个最优解，其中  $o_1, o_2, \dots, o_k$  为停靠的补水点。

设  $g_1$  是离起点最远的并且可以能够到达的点。

因为  $o_2 - o_1 \geq o_2 - g_1$ ，于是我们可以将  $o_1$  用  $g_1$  代替，来得到一个新的解  $G_1$

并且由于  $G$  与  $O$  具有相同数量的点，所以  $G$  也是一个最优解。

因为该问题具有最优子结构，因此可以不断重复下去，找到一个  $G_k$ 。

综上，得证。

## 三、算法复杂度分析

---

显然，该问题算法的时间复杂度为  $O(n)$ 。因为要遍历一遍所有的点。

## 四、源码

```
/*
 * @Description: 算法导论第16章贪心算法16.2-4实现
 * @Author: rainym00d
 * @Github: https://github.com/rainym00d
 * @Date: 2020-11-11 20:38:54
 * @LastEditors: rainym00d
 * @LastEditTime: 2020-11-13 15:25:11
 */
#include <iostream>
#include <cstdlib>
#include <vector>

using namespace std;

vector<int> GetWater(int &m, vector<int> &stops)
{
    vector<int> ans;
    int n = stops.size();
    int already_go = 0;

    // 从1开始, 0是出发点
    for (int i = 1; i < n; i++)
    {
        if (stops[i] > m + already_go)
        {
            // 加入站点
            ans.push_back(i - 1);
            // 增加已经走过的路程
            already_go = stops[i - 1];
        }
    }
    return ans;
}

int main(int argc, char const *argv[])
{
    // 初始化
    int a[] = {0, 30, 40, 60, 100, 130, 170};

    vector<int> stops(a, a + sizeof a / sizeof a[0]);
    int m = 40;

    vector<int> ans = GetWater(m, stops);
```

```

    cout << "一共经过" << ans.size() << "个站点" << endl;
    cout << "站点为: ";
    for (auto i : ans)
    {
        cout << i << " ";
    }
    cout << endl;

    return 0;
}

```

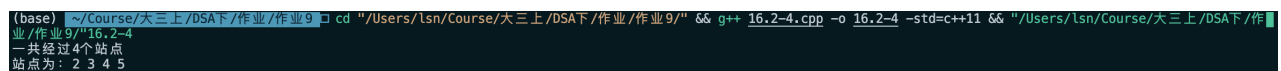
## 五、运行结果截图

输入样例：

$m = 40$

$stops = [0, 30, 40, 60, 100, 130, 170]$

结果



```

(base) ~/course/大三上/DSA下/作业/作业9: cd "/Users/lsn/course/大三上/DSA下/作业/作业9/" && g++ 16.2-4.cpp -o 16.2-4 -std=c++11 && "/Users/lsn/course/大三上/DSA下/作
业/作业9/"16.2-4
一共经过 4 个站点
站点为: 2 3 4 5

```

## 六、问题与总结

总的来说，本次实验代码实现较为简单，没有任何的难度。唯一有难度的地方在于贪心算法的证明上。其中最优子结构的证明是需要好好考虑的一个步骤，因为若没有最优子结构的话，则不可能存在贪心算法。