

实验环境:

```
OS:ubuntu 18.1.04 LTS
compiler:c++11,g++
MakeTool:CMake
```

实验原理

贪心原理,我们要使加油次数最少,那么我们每次加油都尽可能让车的油消耗完.
 那么我们找到当前位置消耗完油的地方之前的最近的加油站.
 由于整个加油站距离起点的位置是递增的,那么我们可以二分查找找到我们要去的加油站.
 这里的二分查找过程,我们做一个简单的优化,如果要找的位置比下一个位置还要近,那么返回-1代表找不到,
 如果找的位置恰好是下一个,那么返回下一个点的位置,
 如果要找的位置比最后一个位置还要远,那么直接返回最后一个位置

实验过程

GasStation.h

```
//
// Created by sumover on 19-4-28.
//

#ifndef ALGORYTHM_DESIGN_GASSTATION_H
#define ALGORYTHM_DESIGN_GASSTATION_H
/**
 * donnot use this function!
 * @param begin
 * @param end
 * @param pos
 * @return
 */
int *_search(int *begin, int *end, int pos) {
    if (pos < *begin) return nullptr;
    if (pos == *begin) return begin;
    if (pos > *(end - 1)) return end - 1;
    int *__begin = begin, __len = (end - begin);
    while (true) {
        if (__len < 2) {
            return __begin;
        } else {
            int __middle = __len / 2;
            if (pos >= *__begin[__middle]) {
                __begin += __middle;
                __len = (__len + 1) / 2;
            } else {
                __len /= 2;
            }
        }
    }
}
```

```

/**
 * count how many times you add oil.
 * @param n
 * @param k
 * @param pos
 * @param cnt
 * @return
 */
int countAddOil(int n, int k, int *pos, int cnt = 1) {
    if (k == 1) return cnt;
    int *next_pos = _search(pos + 1, pos + k, *pos + n);
    if (!next_pos) return -1;
    return countAddOil(n, k - (next_pos - pos), next_pos, cnt + 1);
}

#endif //ALGORYTHM_DESIGN_GASSTATION_H

```

main.cpp

```

#include <bits/stdc++.h>
#include "GasStation.h"

int main(int argc, char **argv) {
    /*
    int *arr = new int[5]{2, 4, 6, 8, 10};
    std::cout << _search(arr, arr + 5, 10) - arr;

    /**/
    int n, k;
    std::cin >> n >> k;
    int *arr = new int[k];
    for (int i = 0; i < k; ++i)
        scanf("%d", arr + i);
    std::cout << countAddOil(n, k, arr) << std::endl;/**/
    return 0;
}

```

运行结果

input:

```

4 5
0 3 4 5 11
3 3
0 3 6

```

output:

```

-1
3

```