

# Homework7

软件13 杨楠

## 1

### a.

证明：从第一行开始，往下逐行选择像素。每一行选了像素后，下一行有2种或者3种选择（所选像素在第一列或者第n列时，下一行的选择是2种），从而可能的接缝数量  $S$  满足  $n \cdot 2^m \leq S \leq n \cdot 3^m$ ，即  $S = \Omega(2^m)$ 。

### b.

令  $s[i, j]$  为选到第  $i$  行第  $j$  个像素的情况下，当前所选像素破坏度总和的最小值。显然有  $s[1, j] = d[1, j], 1 \leq j \leq n$ ，再令  $s[i, 0] = s[i, n+1] = \infty, 1 \leq i \leq m$ ，那么  $s$  的最优子结构满足：

$$s[i, j] = \min\{s[i-1, j-1], s[i-1, j], s[i-1, j+1]\} + d[i, j]$$

整张图片的破坏度最低的接缝，其破坏度即为  $\min\{s[m, j], 1 \leq j \leq n\}$

伪代码如下，返回的数组  $seam[i]$  用于记录第  $i$  行选的是第几列的像素点。

```
GET-SEAM(d, m, n)
    let s[1..m, 0..n+1] be a new table
    let result[1..m, 1..n] be a new table
    for i = 1 to m
        s[i, 0] = s[i, n+1] = inf
    for j = 1 to n
        s[1, j] = d[1, j]
    for i = 2 to m
        for j = 1 to n
            s[i, j] = min(s[i-1, j-1], s[i-1, j], s[i-1, j+1]) + d[i, j]
            p = j-1
            if s[i-1, p] > s[i-1, j]
                p = j
            if s[i-1, p] > s[i-1, j+1]
                p = j+1
            result[i, j] = p

    let seam[1..m] be a new array
    q = 1
    for j = 1 to n
        if s[m, q] > s[m, j]
            q = j
    seam[m] = q
    for i = m to 2
        seam[i-1] = result[i, seam[i]]
    return seam
```

算法的时间复杂度为  $\Theta(m + n + mn + n + m) = \Theta(mn)$

## 2

贪心算法设计如下：

先将活动按照结束时间进行排序。然后从头遍历每个活动。将第一个活动安排在第一个教室。对于每个后面的活动，如果它的开始时间比前一个活动的结束时间晚，那么将其安排在前一个活动使用过的教室。否则，将这个活动安排在一个新的未被使用过的教室。遍历直至结束。

排序可以在  $O(n \lg n)$  时间完成，遍历活动的时间则为  $O(n)$ ，总的时间复杂度为  $O(n \lg n)$ 。

```
ACTIVITY-SELECT(S)
  n = S.length
  sort S by their finish time
  num = 1
  let place[1..m] be a new array
  place[1] = num
  for i = 2 to n
    if start[i] < finish[i-1]
      num = num + 1
  place[i] = num
  return place
```