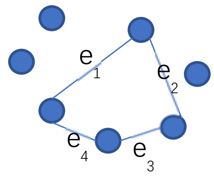
非编程题部分

1. 假定 G 是一个无向图 且 此图中所有边的权值**各不相同**。
假设$(e_1,e_2,\ldots,e_k)$是 G 中的 k 条边，并构成一个简单回路。
求证：这个回路中权值最大的那一条边 一定不在 G 的最小生成树当中。



2. 无向图 G 有 n 个顶点、n+8 条边、且是连通的（假定所有边的权值各不相同）。请你描述一个算法计算 G 的最小生成树，要求时间复杂度为 O(n)。(对这道题，你只需要简述算法正确性不必太严谨的证明)

3. 给定字符串 $T=T_1\cdots T_n$。定义$\pi(j) = \max\{k<j \mid T[1,j]$的长为 k 的前、后缀相等$\}$。
严格的证明： $\{k \mid T[1,j]$的长为 k 的前、后缀相等$\} = \{j, \pi[j], \pi[\pi[j]], \pi[\pi[\pi[j]]], \cdots, 0\}$。

4. 假设 $\mathbf{x}=(x_1,\cdots,x_n)$ 是一个实数序列，并且任意两个元素互不相同。Slide 7 中给出了求 x 的*最长递增子序列*的 $O(n^2)$的算法。在 slide 20 中，利用 AVL 树，我们给出了一个更快的 O(n log n)算法。实际上，对于最长递增子序列问题，存在一个更加简单的 O(n log n)算法，它不需要用到诸如 AVL 树、Splay 树这样的高级数据结构，而是只要用到数组。请设计出这个终极的 O(n log n)算法。要求认真描述清楚算法，并简单扼要的写出证明。

编程题部分

你只需要提交 cpp 代码。要求用 dev c++能编译通过(版本:c++11)
我们将采用机器评测。请务必严格遵守我们对输入输出的格式要求。

5. slide7 上讲了如下的**任务安排问题**。有 n 个任务 1~n。假设同一个时间单位你只能处理一项任务；任务 i 只能在时刻 r_i 后开始处理。($r_1,\cdots r_n$是给定的) 任务 i 需要 p_i 个时间单位才能完成。($p_1,\cdots,p_n$是给定的) 一个任务可分成多次处理（可停下来等之后继续处理）。要设计方案使得$\sum t_i$最小，t_i 为任务 i 被完成的时刻。**请你给出贪心算法的 c++实现**。

6. 输入矩阵 $A=(A_{i,j}:1\le i\le m,1\le j\le n)$。如果 $1\le i_1\le i_2\le m$ 且 $1\le j_1\le j_2\le n$，则 $B=(A_{i,j}: i_1\le i\le i_2,j_1\le j\le j_2)$是 A 的子矩阵。A 的一个子矩阵的和定义为其各元素之和。设 B*是 A 的和最大的子矩阵，请输出 B*的和。

7. 给定无向带权图 G=(V,E)。它有 n 个顶点 m 条边，其中 $n\le 50000, m\le 500000$。顶点从 1~n 编号。各边权值为 1~1000 之间的整数。请用 Dijkstra 算法计算从顶点 1 出发到各顶点的最短路径长度。输入数据保证 G 是连通图，因此顶点 1 可以到其余任何顶点。
说明：请务必使用堆优化——如果你的时间复杂度为 $O(n^2+m)$则会超时而得不到分数。

8. 请用伸展树(见 slide 22)实现如下功能。一开始设集合 S 为空。
   1. **Insert**(x); //往 S 中添加元素 x。（我们保证 x 不在 S 中）
   2. **Delete**(x); //从 S 中删除元素 x。（我们保证 x 已在 S 中）
   3. **Rank**(x); //在 S 中寻找元素 x 并输出 x 的 rank（即 x 是 S 中第几小的元素？）（我们保证 x 在 S 中）
   4. **Get**(int k); //找到 S 中第 k 小元素并输出。 （保证|S|≥k）

每次操作结束时执行 Splay 操作并打印相关信息（见输入输出格式说明）。