Tasks related to trees

Task 1【有向树形图】

任务介绍：slide15 中大致介绍了有向树形图(Arborescence)。给定一个带权有向图 G，已知 Edmond 算法可以计算图 G 的最小代价有向树形图。请你学习这一算法，阅读相关资料，并撰写阅读报告。(要求掌握 O（|E| log|V|）或 O(|E|+|V| log|V|)的实现)

en.wikipedia.org/wiki/Arborescence_(graph_theory)
en.wikipedia.org/wiki/Edmonds%27_algorithm

Task 2【树状数组】

任务介绍：树状数组(Binary Indexed Tree；Fenwick tree)是一种简单的数据结构，它支持两种操作：1. 修改 A_i 的值($1 \le i \le n$)；2. 回答 $A_1+A_2+\ldots+A_i$($1 \le i \le n$)。该数据结构每种操作的时间复杂度都为 O(log n)，而空间复杂度为 O(n)。(回顾一下，我们在课上教授的二叉堆 binary heap 也是类似的支持两种操作：一、修改 A_i; 二、返回 $\min(A_1,\ldots,A_n)$。而且每种操作也是 O(log n)时间。) 请自学树状数组，并攥写阅读报告（要求写明两种操作是如何实现的）。

en.wikipedia.org/wiki/Fenwick_tree
oi-wiki.org/ds/fenwick/

Task 3【最佳判定树】

任务介绍：slide12 中给出了动态规划转移方程计算最佳判定树：

$$f[i][j] = \begin{cases} 0, & i = j; \\ \min\limits_{k}\left(f[i][k] + f[k+1][j] + w_i + \cdots + w_j\right), & i < j. \end{cases}$$

我们知道，用基本的方法来解决上述方程需要 $O(n^3)$的时间。但是，Knuth 发现了一个巧妙的技术，能够在 $O(n^2)$时间解决该方程（而且非常简单）。实际上，有一大类转移方程都可以应用这一技术；简单的说，如果转移方程 f 是 2 维的且满足某种称作“**四边形不等式**”的性质，那么我们可以在 $O(n^2)$时间内解决它。请你学习四边形不等式优化技术，并撰阅读报告。你需要写明该技术适用的范围及原理（它是如何降低复杂度的）并给出适当的分析。

en.wikipedia.org/wiki/Optimal_binary_search_tree
oi-wiki.org/dp/opt/quadrangle/
link.springer.com/article/10.1007/BF00264289 (Knuth 原文)

Task 4【信源编码定理】

任务介绍：字符集 Σ 含有 n 个不同字符，且它们出现的概率分别为 $p_1 \sim p_n$。对 Σ 中的字符进行编码：每个字符用字符集 Σ_2 中的一个*字符串*表示（设字符集 Σ_2 中有 a 个字符）。要求用课上介绍的**前缀编码方式**，即一个字符的编码不能是另一字符的编码的前缀。为何用另一个字符集对 Σ 中的字符进行编码呢？这是由于在信道传输过程中，单位信号只有少数几个选则如'高'、'低'（即 1、0）！前缀编码 S 的**平均码长**$L(S) := \sum_{i=1}^{n} p_i l_i$，其中$l_i$代表第 i 个字符的编码长度。信息论鼻祖香农给出了如下的"**信源编码定理**"：设 S* 为平均码长最小的前缀编码，则$H/\log a \le L(S^*) \le H/\log a + 1$，其中$H = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$——这是分布为$(p_1,\ldots,p_n)$的随机变量的**熵**。请阅读上述定理的完整证明并撰写报告。证明用到了有趣的不等式(Gibbs'和 Kraft's inequality)，你的报告中必须给出它们的证明。

https://en.wikipedia.org/wiki/Shannon%27s_source_coding_theorem
https://en.wikipedia.org/wiki/Entropy_(information_theory)

Task 5【并查集的复杂度】

任务介绍：slide13 中介绍了并查集的运行复杂度，其中有一个定理说到"*Starting from an empty data structure, link-by-rank with path compression performs any intermixed sequence of m ≥ n MAKE-SET, UNION, and FIND operations on a set of n elements in O(m log* n) time.*"请你认真读懂这个定理的证明，并撰写阅读报告。

www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/UnionFind.pdf

Task 6【Aho-Corasick 自动机（AC 自动机）】

任务介绍：有 m 个模式串 $S_1,S_2,\ldots S_m$ 以及一个文本串 T。要找出所有模式串在 T 中的出现(occurrences)。如果用 KMP 算法，这要 $O(|S_1|+\ldots|S_m|+m|T|)$时间。Aho-Corasick 给出了一个更高效的 $O(|S_1|+\ldots|S_m|+|T|+t)$时间算法找出所有出现，其中 t 表示模式串出现的总次数。请阅读 Aho-Corasick 自动机的相关文献，并撰写报告（理解构造 trie、用 bfs 构造 failure 指针、及最后的匹配过程）。

en.wikipedia.org/wiki/Aho%E2%80%93Corasick_algorithm
oi-wiki.org/string/ac-automaton/

更进一步的，请思考如何利用 AC-自动机解决如下问题。（将它的解法作为范例写入你的报告中）www.luogu.com.cn/problem/P2444

Task 7【树的顶点标号】(*)

任务介绍:给一棵 n 个节点的树。用 k 种颜色（1~k）对节点着色（不同顶点允许着相同的着色）满足：对于任意两个着色相同的节点，在连接它们俩的唯一的路径上存在另一个节点它使用的着色更大。k 并不是问题输入；你要找一种着色方案使 k 最小。

该问题存在 O(n log n) 甚至 O(n)时间的算法（且一些推广问题仍能在线性时间内解决）。请你先思考该问题如何解决（提示：贪心）之后阅读以下资料并撰写报告。(需掌握 O(n log n)的算法；无需在报告中介绍 O(n)算法)。 <树的顶点标号解题报告> <Optimal node ranking of trees, Information Processing Letters 1988> < Optimal node ranking of trees in linear time, Information Processing Letters> <Generalized vertex-rankings of trees, Information Processing Letters>

Task 8【Fair split tree】(**)

任务介绍：slide7 讲解了最近点对问题的分治算法。现在我们考虑如下的拓展问题——All Nearest Neighbor Problem：输入平面上的 n 个点 $p_1,\ldots,p_n$，为每一个 p_i 找到距离它最近的$\{p_1,\ldots,p_n\}\backslash\{p_i\}$中的点。这个问题比原最近点对问题更难；实际上，课上所介绍的分治算法并不能解决这个拓展问题！然而这个拓展问题依然存在 O(nlogn)时间的解法——Callahan 和 Kosaraju 在 1995 年给出了这样一个算法，基于 Well-separated pair decomposition 和 Fair split tree。阅读他们的论文中关于上述问题的部分并攥写报告。

<Callahan & Kosaraju, JACM 1995, A Decomposition of Multidimensional Point Sets with applications to k-Nearest-Neighbors and n-Body potential Fields>

Task 9【生成树记数】(**)

任务介绍：课上我们学习了最小生成树，但是关于生成树还有许多有趣的问题。比如，给定无向图 G，如何计算 G 的生成树的个数。这个问题可被完美解决——它存在多项式时间算法。请学习 Kirchhoff's theorem（en.wikipedia.org/wiki/Kirchhoff%27s_theorem），并使用它得到上述问题的多项式时间解法（该定理给出了一个简单的公式，把生成树的计数与一个 矩阵的行列式联系起来了）。

其他说明：你可以先自己思考一下，如果 G 是 n 个点的完全图，G 有多少个生成树呢？该问题有一个非常简单的答案。请阅读：

en.wikipedia.org/wiki/Cayley%27s_formula 以及

en.wikipedia.org/wiki/Pr%C3%BCfer_sequence

Tasks not related to trees

Task 10【最小表达】

任务介绍：slide5 介绍了字符串的 rotation(循环移动后的字符串)。现在请考虑如下问题：给定一个字符串 $S=s_1…s_m$，在 S 的 m 个 rotation 中找到字典序最小的一个（即，找到 S 的**最小表示**）。例如，'abcab'的最小表示为'ababc'。你可以想出 O(m)的算法吗？

Booth 和 Shiloach 分别给出了一个 O(m)时间的算法解决这个问题。请你**任选其中一个**进行学习，并撰写阅读报告。Booth 算法更简单，但是 Shiloach 算法更快（它的隐藏在大 O 记号后面的常数因子更小）。Booth's 算法与 KMP 算法有异曲同工之妙。

en.wikipedia.org/wiki/Lexicographically_minimal_string_rotation

<Booth, Lexicographically least circular substrings, Information Processing Letters 1980>

<Shiloach, Fast canonization of circular strings, Journal of Algorithms, 1981>

Task 11【Lyndon 分解】

任务介绍：Slide5 介绍了 Lyndon word 以及 Lyndon Factorization。Chen–Fox–Lyndon Theorem 指出，任意单词 w 可以被唯一的分解为 $w = w_1 w_2 … w_m$，使得每一个 w_i 都是 Lyndon word，并且 $w_1 \geq w_2 \geq … \geq w_m$。这种分解称作 Lyndon Factorization。J.P. Duval 在 1983 年的论文<Factorizing words over an ordered alphabet, Journal of Algorithms>中给出了一个十分漂亮的线性时间算法计算任何字符串 w 的 Lyndon Factorization，并且给出了这种分解的几个重要的应用。请你阅读该文并撰写阅读报告。

Task 12【无向图的桥】

任务介绍：在无向图中，一条边被称作**桥**如果删去它以后整个图中连通分量的个数会增加；等价的说，一条边是桥如果它不在任何简单回路中。比如说，一棵 n 个节点的树具有 n-1 条边；每一条边都是这个树的一个桥。Robert Tarjan 给出了一个线性时间的算法来找到一个无向图中所有的桥。请学习该算法并撰写报告。此外请完成<算法导论>第三版习题 22-2；答案纳入你的报告中。

en.wikipedia.org/wiki/Bridge_(graph_theory)

<Tarjan, A note on finding the bridges of a graph, Information Processing Letter 1974>

Task 13【均值最小的环】

任务介绍：给定一个带权的有向图 G。如果 $c=<e_1,\ldots,e_k>$构成一个环(loop)，我们定义这个环的均值为$[w(e_1)+\ldots+w(e_k)]/k$，其中 $w(e_j)$表示边 e_j 的权值。请思考如何找出 G 中**均值最小的环**。然后，完成《算法导论(第三版)》习题 24-5。通过这个习题，你将学会 Karp's minimum mean-weight cycle algorithm，它给出了一个 O(|V||E|)的算法来计算均值最小的环。如果该习题某几问你做不出来，可在网络上找到答案。最终，请给出 Karp's 算法的报告。

walkccc.github.io/CLRS/Chap24/Problems/24-5/
courses.csail.mit.edu/6.046/fall01/handouts/ps9sol.pdf

Task 14【中位数确定性算法】

任务介绍：Slide7 中介绍了寻找中位数的一个递归的算法。该算法是一个随机算法。可以证明它的期望运行时间是 O(n)（但是该证明我们未在课堂上给出）。我们将在本课程的后半学期学习快速排序算法，它同样是一个递归算法。上述两个算法共同的一个关键步骤是**选取一个 pivot point**；然后将小于它的数放到靠前的位置；将大于它的数放到靠后的位置。如果 pivot point 选的不好（比如它比其他数都小或者比其他数都大），那么运行时间就会偏多。事实上，如果要给出一个确定性的 O(n)时间的中位数算法，关键点在于选取 pivot point。现有一种称作“**Median of medians**”的方法可用来选择 pivot point。请你学习该方法并撰写阅读报告（应清楚说明 median of median 的核心思想、证明思路、以及如何用它作为一个模块来设计确定性的 O(n)时间的中位数算法）。

en.wikipedia.org/wiki/Median_of_medians
brilliant.org/wiki/median-finding-algorithm/
rcoh.me/posts/linear-time-median-finding/
www.cs.cmu.edu/~avrim/451f11/lectures/lect0908.pdf

Task 15【Randomized incremental algorithm】

任务介绍：在这个任务中，你需要系统的学习 randomized incremental algorithm 这一算法，通过阅读书籍《Computational Geometry:Algorithms and applications(3 edition)》的第 4 章。尤其是 4.1-4.4 以及 4.7。读懂这些章节并撰写阅读报告。(给定平面上若干点，该算法可在线性时间内计算最小的圆包含这些点。)

Task 16【Chan's convex hull algorithm】(*)

任务介绍：Slide2 介绍了计算凸包的 Graham-scan 算法；分治算法也可计算凸包；它们的时间复杂度都是 O(n log n)。实际上还有许多其他算法可计算凸包，如 Jarvis-March 算法，它仅需要 O(nh)时间，其中 h 为凸包上的点数。但是，当 h 超过 log n 这个量级时 Jarvis-March 算法的效率不如前面两个算法。1996 年华人科学家 T. M. Chan 给出了一个终极的凸包算法，时间复杂度 O(n log h)。比 O(n log n)与 O(nh)都好！且该算法可以容易的推广到 3 维情形（但是在本任务中，你只需考虑 2 维情形，即假定所有的点都在平面上）。请你学习 Chan's algorithm 并攥写阅读报告。

en.wikipedia.org/wiki/Chan%27s_algorithm
en.wikipedia.org/wiki/Gift_wrapping_algorithm
<T.M. Chan, Optimal output-sensitive convex hull algorithms in two and three dimensions, Discrete and Computational Geometry 1996>

Task 17【Maximum subarray query】(**)

- 任务介绍：slide7 介绍了"和最大的连续子序列问题"之后谈到一个拓展问题：寻找 k 个连续子序列，它们彼此不相交，且它们的和最大。Homework 1 中讨论了这一问题的动态规划解法。但是该算法的复杂度很高。实际上，存在 O(n)的算法来解决这一拓展问题。（通过一系列转换）拓展问题可以转化为回答 O(n)个如下查询：给定（i,j），需要返回在 a[i]...a[j]中和最大的连续子序列是哪一段。这种查询在 O(n)时间的预处理后，可以在 O(1)时间回答。请阅读论文<Chen and Chao, On the range maximum-sum segment query problem>，了解这是如何实现的，并攥写阅读报告。

en.wikipedia.org/wiki/Maximum_subarray_problem

Task 18【分治算法进阶】(**)

任务介绍：首先，考虑如下问题。给定正整数 K 和树 T，树上每条边有一个距离（大于 0），你需要统计有多少个节点对，它们之间的距离不超过 K （这是 slide7 的课后思考题）。请你思考如何解决这一问题（提示：可使用分治思想；该问题的答案在附件中提供）。然后请你学习<algorithm design>5.6 章<Convolutions and the Fast Fourier Transform>。最后，请你写一个阅读报告：简要描述距离统计问题以及 Fast Fourier Transform 问题的分治算法。