Data Structure and Algorithm Course Design Prototype

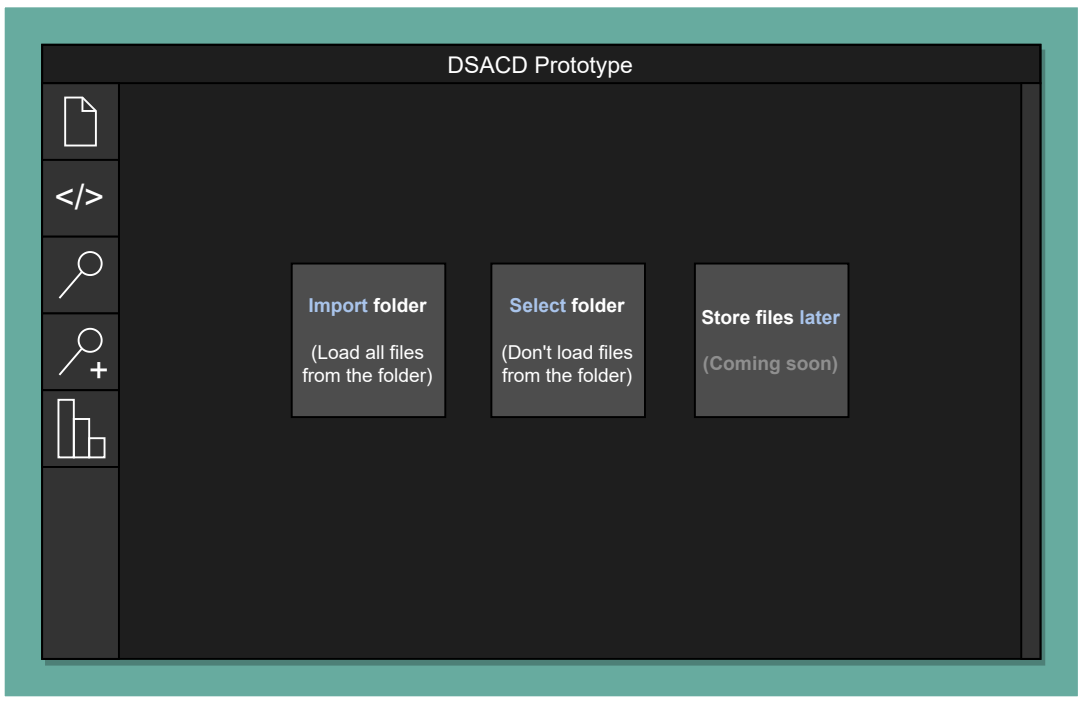
孙天天 19071110

整体设计

使用技术栈：
* Vue+AntDesign (HTML+CSS+JS)
* 除了组件库不使用其他库
* 使用同义词数据库（json文件存储）联想同义词

自定义扩展功能：
* 树状文件系统
* 高亮、自动选中检索/替换内容，一键全部替换

达成题目要求：
* 扩展要求(1) “界面设计的优化。”
* 扩展要求(5) “可以自行根据本题目程序的实际应用情况，扩展功能。”



Start - 开始界面

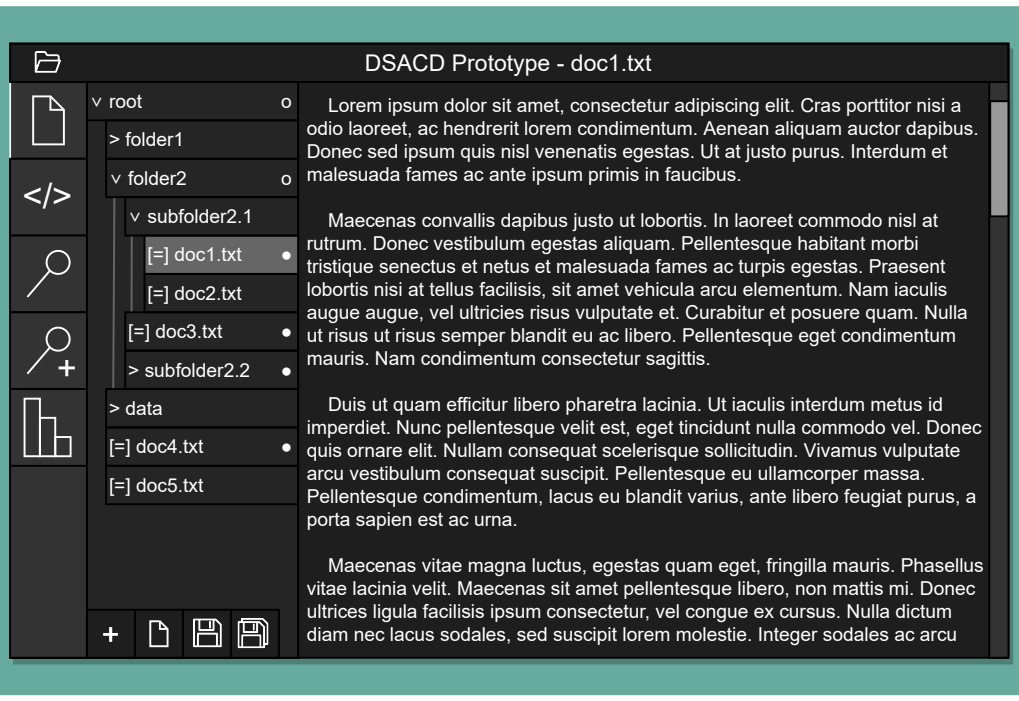
选择一种存储文件方式：
* 载入目录：将目录所有文件递归载入
* 选择目录：选择存放文件的目录但不载入已有文件

Editor - 文本编辑器

文本编辑器：
* 实现基础的文本增添、删除功能
* 使用textarea实现

左上角文件夹按钮：
* 回到开始页面，若有文件未保存会发出提醒

达成题目要求：
* 基础要求(1) “设计图形界面，可以实现英文文章的编辑和检索功能”



File - 文件侧栏

侧栏树状显示目录与文件：
* 展开/收起文件夹
* 选择当前编辑的文件
* 提示尚未保存的文件
（目前显示的最底层元素显示实心圆，其父元素显示空心圆）

下方四个按钮：
* 创建文件：在当前选中目录/选中文件同目录下创建新文件
* 打开文件：同样创建文件，但内容初始化为选中的文件
* 保存当前文件
* 保存全部文件

达成题目需求：
* 基础要求(2)/1 “创建新文件；打开文件；保存文件。”

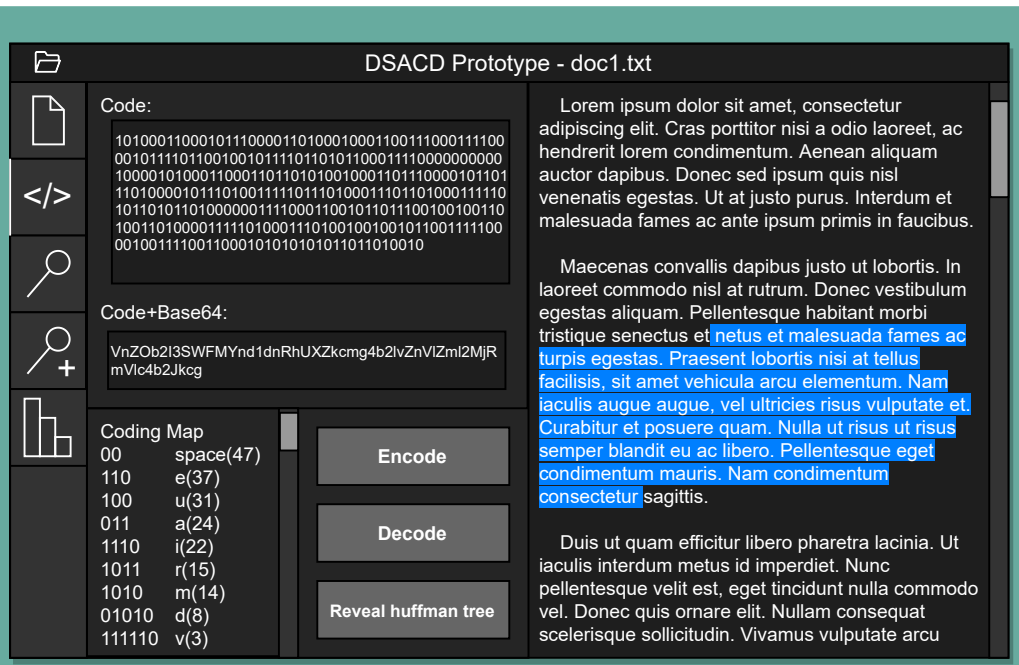
数据结构：树
* 存储形式：对象作为结点+指针
* 载入文件夹：从根递归地创建树
* 选中/修改/保存某个文件：操作指定结点
* 创建文件/打开文件：为树创建新节点
* 显示未保存文件：找某节点所有父元素
* 树的可视化：由组件库实现

Search/Replace - 检索替换侧栏

检索/替换侧栏：
* 展开/收起替换框=检索/替换切换
* 点按钮检索/替换（单个或所有）当前文件中文本
* 列出所有检索结果，高亮检索效果，预览替换效果
* 点击某个结果，右侧选中对应文本

达成题目需求：
* 基础要求(2)/2 “查找：输入单词在当前打开的文档中进行查找，并将 结果显示在界面中。”
* 基础要求(2)/3 “替换：将文章中给定的单词替换为另外一个单词，再保存等。”

算法：KMP
* 每次搜索时对当前文本进行KMP匹配运算
* 列出所有匹配结果（向前寻找3个单词或指定最大字符数）
* 列出结果中关键部分使用特殊格式
* 默认选中第一个检索结果，点击某个时跳转至那一个



Encode/Decode - 编码解码侧栏

编解码侧栏：
* Code栏展示/输入编码结果
* Code+Base64栏展示/输入编码的Base64结果
* Coding Map栏展示编码表
* Encode：将右侧选中内容哈夫曼编码生成到左侧
* Decode：将左侧输入编码生成到右侧
（替换选中内容或增加到光标位置）

达成题目需求：
* 基础要求(3) “对于给定的文章片段（30<单词数量<100），统计该片段中每个字符出现的次数，然后以它们作为权值，对每一个字符进行编码，编码完成后再对其编码进行译码。在图形界面中演示该过程。”
* 拓展需求(2) “对于编码与译码过程，可以自行设计其他算法。” --Base64（不太确定十分符合要求）

算法：桶排序+哈夫曼编码
* 统计词频使用桶排序
* 编码时用词频构建哈夫曼树
* 编码表为哈夫曼树前序遍历
* 译码时用编码表构建哈夫曼树

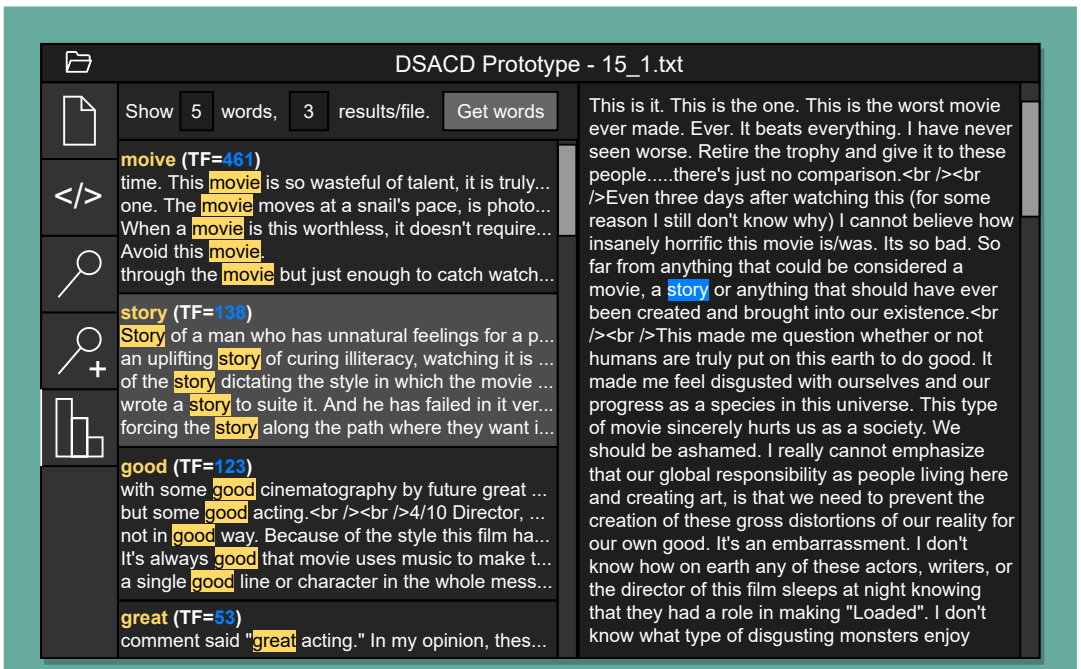
Search+ - 多文件检索侧栏

多文件检索侧栏：
* 按下按钮对多文件检索，并将结果列在下方
* 每条结果包括文件路径、检索结果，高亮所有匹配结果
* 选中某条结果自动打开对应文件，光标选中第一个词
* 支持逻辑表达式
* 支持相近词检索

达成题目需求：
* 基础要求(5) “对于给定的多篇文章构成的文档集中，建立倒排索引，实现按照关键词的检索（包括单个关键词；或者2个及以上关键词的联合检索，要求检索结果中同时出现所有的关键词），并在界面中显示检索的结果（如：关键词出现的文档编号以及所在的句子片段，可以将关键词高亮显示）”
* 拓展要求(3) “高级检索，采用逻辑表达式来表示检索需求，例如：(关键词A) AND (关键词B) OR (关键词C) AND (!关键词D)”
* 拓展要求(4) “优化检索，对于检索结果的相关性排序，例如：包含关键词的数量等信息为依据。” -- TFIDF算法+相关词联想

算法：二级倒排索引
* 一级索引：统计每个文件中各个单词出现的位置
* 二级索引：记录每个单词在各个文件中出现的词频
* 逻辑表达式检索
  * 每个文档单独搜索求出集合，将逻辑运算转换为集合运算
  * 选出计算后符合条件的文档集合，再对文档进行排序
* 相关性检索
  * 使用TF*IDF数值代替原本的TF数值进行逻辑表达式运算
  * 联想同义词参与检索（仅考虑相关性给定数量的同义词）

算法：中缀表达式求值
* 先将中缀表达式转为后缀（使用堆栈）
* 再对每个文件进行后缀表达式求值（使用堆栈）



Frequency - 词频边栏

词频排序：
* 实现按TF检索最高词频，可设置显示条数
* 选中指定条目在右侧显示第一个匹配的文档，自动标注第一个匹配的词

达成题目要求：
* 基础要求(4) “对于给定的多篇文章构成的文档集中，统计不同词汇的出现频率，并进行排序，在界面中显示TOP 30的排序结果。”

算法：二级倒排索引
* 同多文件检索栏使用的二级倒排索引
* 二级索引统计单词出现总次数，并进行排序