Дерево Ван Эмде Боса.

Что нужно было сделать:

- Реализация **кучи** Ван Эмде Боса. У нее должны быть методы Add(x, C), Delete(x, C), ExtractMin, где C - ее верхняя граница (все элементы в куче по построению лежат в отрезке [0; C]).
- Реализация Бинарной Кучи с методами Add(x), ExtractMin. Нужна для сравнения по времени работы.
- Протестировать Ван Эмде Боса.
- Сравнить время работы Van Emde Boas Heap, Binary Heap, priority_queue (std).

Что сделала:

- Реализация Ван Эмде Боса есть *(boas.cpp, boas.h)*
- Реализация Бинарной Кучи есть *(binary.cpp, binary.h)*
- Генератор тестов для Ван Эмде Боса есть *(boas_test.cpp)*
- Написан генератор тестов для сравнения времени работы и само сравнение *(pq_bin_boas.cpp, main.cpp)*

Описание файлов.

boas.cpp.

Тут реализованы методы класса VAB_Heap. Особенность: методам Add и Delete надо передавать верхнюю границу промежутка, на котором работает куча. Наверное, можно было как-то задать дефолтное значение, но я не справилась с этим.

binary.cpp.

Реализация методов класса Bin_Heap. Ничего особенного.

boas_test.cpp.

Генератор тестов для тестирования кучи Ван Эмде Боса. Структура выходного файла boas_test.in. число ntests в первой строчке - количество тестов. Далее ntests блоков, в каждом из которых на первой строке число n - количество операций в тесте, далее n строк с описанием операций.

Каждая из n строк состоит из буквы A (Add), D (Delete) или E (ExtractMin) и числа x. Для добавления и удаления x - число, которое мы хотим удалить/добавить, для изъятия минимального - ответ, который мы должны получить.

pq_bin_boas.cpp.

Генератор тестов для сравнения по времени работы. Формат выходного файла pq_bin_boas.in тот же, что и в boas_test.in, но только там нет операций Delete (т.к. их нет в очереди с приоритетом и бинарной куче) и после символа E не идет число.

main.cpp.

Первая часть программы — проверка корректности написанной кучи Ван Эмде Боса. Результатом ее работы быть файл Boas_results.txt, в котором столько же блоков, сколько было тестов, в каждом из которых на каждую операцию ExtractMin есть строка с выданным моей куче ответом и правильный ответом (его мы взяли из файла boas_test.in).

Вторая часть — сравнение времени работы трех структур. Результат ее работы — файл Compare_results.txt, где столько же блоков, столько было тестов, в каждом из которых три числа - время работы VAB, Bin и prior.