Data Structures - Assignment no. 8

Date due: not for submission

Remarks:

- For every algorithmic problem, try to find the most efficient solution. Analyze the complexity of every algorithm you give, and in case it is not trivial, prove its correctness.
- Unless you are explicitly asked otherwise, please describe algorithms only to the extent which makes them clear to a reasonable reader. You may use plain language, or pseudo-code, whichever is more convenient.

1. Implement a stack using arrays, such that the amortized complexity of each operation is constant (that is, show both an implementation and an appropriate potential function). The size of the arraty should, at all times, be linear in the number of elements in the stack.
2. Implement a queue under the same conditions as the previous question.
3. A med-queue is a data-structures that supports the operations of a queue, and in addition
   - **insert_med**(x) - inserts an element to the middle of the queue. If there are n elements in the queue, x is inserted behind the $\lfloor n/2 \rfloor$'th element, counting from the head of the queue.
   - **delete_med** - deletes the $\lceil n/2 \rceil$'s element from the queue.

   Implement a med-queue using only stacks and $O(1)$ additional memory. The amortized time of each operation must be constant.
4. Give a tight upper-bound on the height of a tree within a binomial heap that contains n elements.