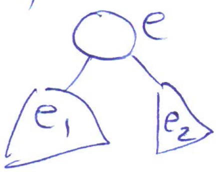
Applications of Priority Queues / Heaps

1. Huffman coding:

   while there is more than ~~1~~ element:

   - find e_1 and e_2 that are least frequent among all elements
   - Merge e_1 & e_2 into a single tree, creating a new element e, with $f(e) = f(e_1) + f(e_2)$

   Output coding tree.

2. Prim's algorithm for MST
3. Dijkstra's algorithm for shortest paths
4. Finding k-best elements of a stream (External k^{th} smallest element problem)
5. Process scheduling, interrupt handling in operating systems
6. Heap sort algorithm
7. Heuristics for Bin packing problem
8. Discrete event simulation, games
   - Event generators
   - Events have to be processed in temporal order
9. A^* search in AI
10. Reducing round-off error in numerical computations:

    given an array of reals, $A[1..n)$, how do you calculate $s = \sum_{i=1}^{n} A[i]$, such that $|S_{calculated} - S_{actual}|$ is a minimum?

    - Best algorithm is similar to Huffman coding algorithm's strategy.
11. Merge sort using k-way merge.
12. Find perfect powers (numbers of the form a^b) in increasing order.

    Find numbers of the form $p_1^i \, p_2^j \, p_3^k$, in sorted order, for primes p_1, p_2, p_3.
13. Pairing buy/sell orders in the stock market by market makers