CS 522: Data Structures and Algorithms II
Extra Credit 1

Dustin Ingram

February 12, 2013

1. **17-1 Solution:**

   (a) The bit-reversal permutation algorithm:

Bit-Reversal-Permutation(A)

```
for i ∈ 0 ... n − 1 do
  if rev_k(i) > i then
    swap(A[i], A[rev_k(i)])
```

   (b) This is nearly identical to a binary counter. The Bit-Reversed-Increment procedure simply flips every bit that is zero to a one and vice versa. The amortized analysis is the same and allows for the bit-reversal permutation of an n-element array in $O(n)$ time.

   (c) Yes, it is possible, and is again similar to the binary counter. We would use the accounting method to make every shift "pay forward" for a future, larger shift.

2. **26-3 Solution:**

   (a) Because the cut (S, T) is a finite capacity cut, only edges of the form $s \rightarrow A_i$ and $J_i \rightarrow t$ cross the cut, and since none of these edges can be infinite (a job cannot have infinite revenue or infinite cost), then if $J_i \in T$ then $A_k \in T$ for each $A_k \in R_i$.

   (b) The maximum net revenue, where c_G is the capacity of the minimum cut of G and corresponds to the cost per expert, is:

$$\left(\sum_{i=1}^{m} p_i \right) - c_G$$

   (c) We will use a Max-Flow algorithm to find the maximum flow of the described flow network G. For every edge (s, A_i) which the algorithm selects, we hire expert A_i and for every edge (J_i, t) which the algorithm selects, we accept job J_i. If we use the optimal max-flow algorithm, the complexity is $O(E \cdot f)$, where E is the number of edges, and f is the max flow. In this case, the maximum number of edges is:

$$E = m + mn + n$$

Here, $f = r$. Thus the complexity of this algorithm is:

$$O(mnr)$$