# Problem 1

a) $S$ is the sum of all object sizes. Each object $a_i$ must have a size $0 < s_i < 1$ and each bin can hold up to a unit of size. By way of contradiction lets assume that the number of bins is less than $\lceil S \rceil$ then by the pigeonhole principle there must exist at least one bin that is holding more than a unit of size $\Rightarrow\Leftarrow$ because we stated that a bin can hold at most one unit of size thus the number of bins must be greater than or equal to $\lceil S \rceil$.

b) Lets prove this inductively.

Base case: trivially if there are no buckets, then at least one of them is not half full.

Lets assume that we now have $k+1$ buckets, where $k$ are more than half full and the last one is less than half full. Now if the next element we add has size $s_i$, we need to show that the invariant is maintained.

if $s_i \leq 0.5$ then we put the element into the last bucket (because it was less than halfway full). After adding the element to the last bucket we either have that the last bucket is still less than halfway full, in which case we still have at most one bucket that is less than halfway full, or the bucket is more than halfway full in which case we have 0 elements that are less than halfway full.

if $s_i > 0.5$ then the element may or may not fit into bucket $k+1$.

if it fits then we now have 0 buckets less than halfway full.

if it doesnt fit then we create bucket $k+2$, which will be more than halfway full leaving $k+1$ to be the only bucket that is not more than halfway full.

In all cases the invariant is maintained and so the firt-fit algorithm ensures that at most 1 bucket is less than half full.

c) We proved that every bucket except the last one are at least halfway full in the first fit algorithm. In the worst case, if there are $k+1$ buckets $k$ of them have "mass" $0.5+\varepsilon$. And 1 bucket would be filled $X$ amount where $X < 0.5$. Then we could say that $S = \frac{k}{2} + \sum \varepsilon_i + X$. As $\varepsilon_i \to 0$ $\sum_i \varepsilon_i \to 0$, so the optimal algorithm could have used $\frac{k}{2}+1$ by putting all of these left overs in the bucket with $X$ amount. Thus because first fit used $k+1$ buckets the competitive ratio is $\frac{k+1}{\frac{k}{2}+1} = \frac{2k+2}{k+2} \leq 2$. Thus first fit is a 2-competitive algorithm.

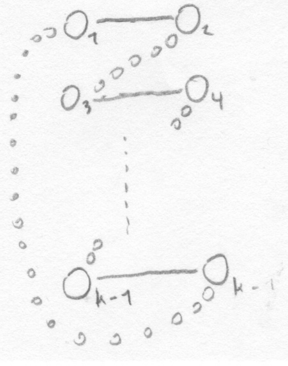
# Problem 2

a) Given an MST T of H, we have that the lower bound on the optimal tour from any given node in T is the sum of all the edges in the tree, that is we manage to touch each edge only once. So we have that $OPT \geq \sum w(e) \ \forall e \in T$.

If we perform a Depth first traversal starting from any node in T we proceed to perform a metric TSP traversal that touches every edge twice think about what happens when the traversal hits a leaf, it has to traverse out on the edge it came in from, and this generalizes to the rest of the nodes. That is the cost of this walk is twice the cost of all the edges in T. And because the OPT walk is at least the cost of all the edges in T, then our depth first traversal costs at most two times what the OPT traversal costs.

b) In every undirected graph every edge connects two vertices. Thus the sum D of degrees of all nodes is an even number, equal to exactly two times the number of edges. The parity of this sum is not affected by the number of nodes with even degree. On the other side if there is an odd number of nodes with odd degree then D would be odd $\Rightarrow\Leftarrow$ because we stated D is always even. Thus it must be that there is always an even number of nodes with odd degrees in an undirected graph.

c) consider the set O of odd degree nodes of in T. Because $O \subseteq$ Nodes of T then it must be that a tour over $O \leq OPT$. A tour over Nodes O could be built with the union of two perfect matchings of O. Since the graph H is complete consider the following arrangement over O




where the two perfect matchings are
$M_1$ with — edges
$M_2$ with oooo edges.

Note that $M_1$ and $M_2$ are sets of edges. while M is the cost of the minimum perfect matching.

The union of the edges in $M_1$ and $M_2$ yields a Tour of the odd degree nodes thus.

$\sum_{e \in M_1} w(e) + \sum_{e \in M_2} w(e) \leq OPT$. WLOG say $\sum_{e \in M_1} w(e) \leq \sum_{e \in M_2} w(e)$. Therefore, it must be that $\sum_{e \in M_1} w(e) \leq \frac{1}{2} OPT$, because if not, then $\sum_{e \in M_1} w(e) + \sum_{e \in M_2} w(e) \geq OPT$.

Since $\sum_{M} M(e) \leq \sum_{M_1} w(e)$ by definition, then it must be that

$\sum_{M} M(e) \frac{1}{2} OPT$

d) Notice that by adding the edges of the perfect matching to the set of even degree nodes we have that all those nodes are now even degree because we added exactly one more edge to every node. Thus now the union of the edges in T and M makeup a multigraph where every node is even. Thus this graph now has an eulerian tour, which will count ano edge only once, and visit all nodes. Thus we have a TSP tour that visits and edge only once. Thus because in part a we said that $T_c = \sum_{e \in T} w(e) \leq OPT$ and from part c we had that $M \leq \frac{1}{2} OPT$ then because the cost of the eulerian tour is $T_c + M \leq OPT + \frac{1}{2} OPT$. And thus we can create an approxiaton of a TSP tour over the original graph that is at most $\frac{3}{2} OPT$

Problem 3.

Let us have a set $P$ of proteins. where $|P| = n$.
For each protein $p_i$ in $P$ create a set $S_i$ where $\{q \in S_i : d(p_i, q) \leq \Delta\}$.
Now run the greedy algorithm of set cover on all sets $S_i$. If the greedy algorithm picks set $S_i$, then $p_i$ is in the representative set of $P$.

Now we need to show that a solution of the protein problem gives us a solution of the setcover problem of the same size and a solution of the set cover problem gives us a solution of the protein problem of the same size.

A solution of the protein problem consists of a set of proteins $Q$ such that for every protein $p \in P$, there is some protein $q \in Q$, st $d(p, q) \leq \Delta$. therefore because for every protein $q_i$ we have set $S_i$ then it must be that $\bigcup S_i$ for all $q_i$ in $Q$ gives us the set $P$. That is the mapping of $q_i$ to $S_i$ $\forall q_i$ gives us the set cover of $P$.
More over, because every $q_i$ map's to $S_i$, and $S_i$ corresponds only to $q_i$ the mapping is a bijection, and the set $Q$ and the set cover are the same size.

A solution to the set cover problem is the set $SC$ for which $\bigcup$ of all $S_i$ in $SC$ is $P$. Therefore, because each $S_i$ corresponds to a $q_i$ and every $t_j$ in $S_i$ has $d(q_i, t_j) \leq \Delta$ by design, then the set $Q$ of all $q_i$ is a representative set of $P$. Lastly, because of the bijection proved above, these two sets $(SC, Q)$ are the same size.
WLOG, these two sets $(SC, Q)$ are the same size.

Altogether because the mapping proposed above create solutions of the same size, and we proved in class that we can solve set cover with a $O(\log n)$ approximation, the above algorithm gives a $O(\log n)$ approximation for the representative set of $P$.