1. (4 points) Suppose you want to satisfy many clauses of an instance of 3-SAT. Show that for any instance of 3-SAT there is an interpretation that satisfies at least $7/8$ of the clauses. That is, if the instance has $m$ 3-clauses, then there is an interpretation that satisfies $\left\lceil \frac{7m}{8} \right\rceil$ of them.

2. (20 points) Assume that $n$ people want to schedule intervals of time to use a computer, where the computer can not be shared at any time among two or more people. Thus, the people's requests are $\rho = \{(s_1, f_1), \ldots, (s_n, f_n)\}$, $s_i < f_i$, $1 \leq i \leq n$. Intuitively, if $(s_i, f_i) = (10, 22)$, this means that person $i$ wants to use the computer for the time interval running from time $s_i=10$ to time $f_i=22$. As in life, we assume that people's wants/needs are inflexible; they are not willing to negotiate. Requests $(s_i, f_i)$ and $(s_j, f_j)$ are compatible if $f_i \leq s_j$ or $f_j \leq s_i$, and a set of requests is feasible if they are pairwise compatible. An optimal schedule is a maximum feasible subset of $\rho$.

a) Consider the following algorithm:

\begin{verbatim}
GREEDY1
\Gamma \leftarrow \emptyset
while there are requests in $\rho$ which are compatible with each request in $\Gamma$ do
    Select $(s_i, f_i)$ which is compatible with each request in $\Gamma$ and minimizes $f_i - s_i$
    $\Gamma \leftarrow \Gamma \cup \{(s_i, f_i)\}$
    $\rho \leftarrow \rho - \{(s_i, f_i)\}$
end while
\end{verbatim}

Clearly this algorithm computes a feasible set of requests $\Gamma$. Either prove that GREEDY1 always produces an optimal schedule, or show that it can fail.
Consider the following algorithm:

**GREEDY2**

\[ \Gamma \leftarrow \emptyset \]

**while** there are requests in \( \rho \) which are compatible with each request in \( \Gamma \) **do**

Select \((s_i, f_i)\) which is compatible with each request in \( \Gamma \) and minimizes \( s_i \)

\[ \Gamma \leftarrow \Gamma \cup \{(s_i, f_i)\} \]

\[ \rho \leftarrow \rho - \{(s_i, f_i)\} \]

Clearly this algorithm computes a feasible set of requests \( \Gamma \). Either prove that **GREEDY2** always produces an optimal schedule, or show that it can fail.

c Consider the following algorithm:

**GREEDY3**

\[ \Gamma \leftarrow \emptyset \]

**while** there are requests in \( \rho \) which are compatible with each request in \( \Gamma \) **do**

Select \((s_i, f_i)\) which is compatible with each request in \( \Gamma \) and minimizes \( f_i \)

\[ \Gamma \leftarrow \Gamma \cup \{(s_i, f_i)\} \]

\[ \rho \leftarrow \rho - \{(s_i, f_i)\} \]

Clearly this algorithm computes a feasible set of requests \( \Gamma \). Either prove that **GREEDY3** always produces an optimal schedule, or show that it can fail.

In real life scheduling problems, some jobs are more important than others. In the *optimal weighted scheduling problem*, the input is \( \rho = \{(w_1, s_1, f_1), \ldots, (w_n, s_n, f_n)\} \), \( w_i \geq 0 \) for all \( 1 \leq i \leq n \), and an optimal schedule is a feasible schedule \( \Gamma_w = \{(w_1, s_1, f_1), \ldots, (w_n, s_n, f_n)\} \) which maximizes \( \sum_{1 \leq i \leq j} w_j \) over all feasible schedules.

That is, it is a feasible schedule which maximizes the sum of the weights of the scheduled requests. You may assume that \( f_1 \leq f_2 \leq \ldots \leq f_n \).

d Describe a linear (in \( n \)) time algorithm to compute a solution to the *optimal weighted scheduling problem*. 
C.S.524
SOLUTION FOR H.W. #4

1. With each of the \( m \) 3-clauses we associate a random variable \( X_i \), \( 1 \leq i \leq m \) which is 1 if an interpretation satisfies the \( i^{th} \) clause and 0 otherwise. For a random interpretation (which chooses values 0 or 1 with equal probability for each variable), the expected value of \( X_i \) is \( 7/8 \). We note that \( \sum_{1 \leq i \leq m} X_i \) is the number of 3-clauses which are satisfied by an interpretation. By Linearity of Expectation,

\[
E[X] = E\left[ \sum_{1 \leq i \leq m} X_i \right] = \sum_{1 \leq i \leq m} E[X_i] = \sum_{1 \leq i \leq m} 7/8 = 7m/8 ,
\]

and there must be some interpretation which satisfies at least this many clauses. Because a clause is either satisfied or not satisfied, there must be an interpretation which satisfies at least \( \lceil 7m/8 \rceil \) 3-clauses.

2. \( \text{a GREEDY1 fails on } \rho = \{(0,5),(4,6),(5,10)\} \). It produces the schedule \( \{(4, 6)\} \), but the optimal schedule is \( \{(0, 5), (5, 10)\} \).

\( \text{b GREEDY2 fails on } \rho = \{(0,10),(1,2),(3,4)\} \). It produces the schedule \( \{(0, 10)\} \), but the optimal schedule is \( \{(1, 2), (3, 4)\} \).

\( \text{c GREEDY3 always succeeds. To see this, let } \Gamma = \{(s_i, f_i), \ldots, (s_l, f_l)\} \) be the schedule constructed by GREEDY3, and assume that it's ordered by the finish times, that is, \( f_k \leq f_{k+1} \) for \( 1 \leq k < l \). Let \( \Pi = \{(s_j, f_j), \ldots, (s_m, f_m)\} \) be any other set of intervals ordered by the finish times. We prove two claims.

\( \text{Claim 1: } f_i \leq f_j \) for \( 1 \leq r \leq l \).

\( \text{Proof of Claim 1: The Claim is clearly true for } r=1. \) Assume that \( f_{i+1} \leq f_{j+1} \). When \( (s_{i+1}, f_{i+1}) \) is added to \( \Gamma \), both \( (s_i, f_i) \) and \( (s_j, f_j) \) are still in \( \Gamma \). Since GREEDY3 chooses \( (s_i, f_i) \) to add to \( \Gamma \), it follows that \( f_i \leq f_j \), establishing the Claim.

\( \text{Claim 2: } \Gamma \) is optimal.

\( \text{Proof of Claim 2: Assume } \Gamma \) is not optimal. Then \( m>l \). By Claim 1, \( f_i \leq f_j \). But because \( \Pi \) is feasible, \( f_j \leq s_{j+1} \). But this is impossible because GREEDY 3 would have added \( (s_{j+1}, f_{j+1}) \) to \( \Gamma \), which yields a contradiction.
For $\rho_w = \{(w_1, s_1, f_1), \ldots, (w_n, s_n, f_n)\}$, we let $\Phi(\rho_w)$ denote the maximum $\sum_{1 \leq k \leq j} w_k$ over all feasible schedules, and we let $\Delta(\rho_w)$ denote $\rho_w$ with $(w_n, s_n, f_n)$ removed as well as all triples $(w_i, s_i, f_i)$ such that $f_i > s_n$, that is, all triples $(w_i, s_i, f_i)$ which are not compatible with $(w_n, s_n, f_n)$. To derive $\Phi(\rho_w)$, we note that either $(w_n, s_n, f_n)$ is included in an optimal schedule, in which case $\Phi(\rho_w) = w_n + \Phi(\Delta(\rho_w))$ or it is not, in which case $\Phi(\rho_w) = \Phi(\rho_w - \{(w_n, s_n, f_n)\})$.

$$\Phi(\rho_w) = \begin{cases} 0, & \text{if } \rho_w = \emptyset \\ \max\left\{w_n + \Phi(\Delta(\rho_{w'})), \Phi(\rho_w - \{(w_n, s_n, f_n)\})\right\}, & \text{otherwise} \end{cases}$$

As a program, having precomputed $\Delta(i)$ for all $i$,

$\Phi[0] \leftarrow 0$

for $i \leftarrow 1$ to $n$ do

$\Phi[i] \leftarrow \max\left(w_n + \Phi[\Delta(i)], \Phi[i-1]\right)$