**first name** **last name** **id#**

**30 min** **30 marks** **closed book** **no devices** **2 pages** 

1. [5+5 marks]

Here is output from the dynamic-program-by-weight (DPW) 0-1 knapsack algorithm with $n = 5$, $W = 18$. Fill in the missing entries (last two columns, last four rows).

```
val      [7,  5,  7,  7, 12]
wt       [7,  5,  8,  6, 10]
 0   0   0   0   0   0   0
 1   0   0   0   0   0   0
 2   0   0   0   0   0   0
 3   0   0   0   0   0   0
 4   0   0   0   0   0   0
 5   0   0   5   5   5   5
 6   0   0   5   5   7   7
 7   0   7   7   7   7   7
 8   0   7   7   7   7   7
 9   0   7   7   7   7   7
10   0   7   7   7   7  12
11   0   7   7   7  12  12
12   0   7  12  12  12  12
13   0   7  12  12  14  14
14   0   7  12  12  14  14
15   0   7  12  14  __  __
16   0   7  12  14  __  __
17   0   7  12  14  __  __
18   0   7  12  14  __  __
```

Here is python code for DPW. It creates the table K shown at left. Fill in the blanks.

```
def knapDP(val, wt, W):

  n = len(val)

  K = [[0 for j in range(n+1)] for w in range(W+1)]

  for j in range(1,n+1):

    for w in range(W+1):

      K[w][j] = _____________________ if w < wt[j-1] \

        else max(_____________________, ______________________)
```

2. [5 marks] Let $S(n)$ be $\{1, 2, \ldots, n\}$. Let $P(n)$ be the set of all subsets of $S(n)$. Let $f(n)$ be the sum, over all $p$ in $P$, of the size of $p$. E.g. $P(3) = \{\{\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$, so $f(3) = 0+1+1+1+2+2+2+3 = 12$. **Claim.** $f(n) = n \times 2^{n-1}$**. Proof of Claim.** Assume $n \geq 1$. Consider the $2^n$ subsets of an $n$-set. If we pair each subset $S$ with its complement $T$ (the set of elements not in $S$), we see that there are exactly $2^n/2 = 2^{n-1}$ pairs of subsets. E.g. **give the four pairs of subsets of** $P(3)$:

_________________ _________________ _________________ _________________

So $f(n)$ (the sum of the sizes of the $2^n$ subsets) equals the sum, over all pairs, of the sizes of the two sets in the pair. **Finish the proof of the claim:**

3. [7 marks] *Set cover* is the problem with these instances and query: *Instance.* A set $U$, a set $S = \{S_1, S_2, \ldots, S_t\}$ of subsets of $U$, and an integer $k$. *Query.* Is there a subset $C$ of $S$ of size $k$ whose union equals $U$?

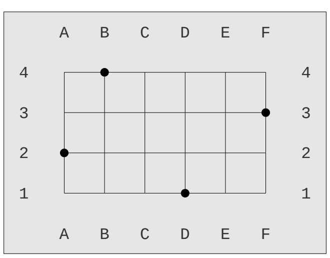
   a) In the set cover instance below, $U = \{0, 1, \ldots, 8\}$ and $S = \{$S0, S1, …, S10$\}$. Does this instance have a cover of size 6? **(circle one)** yes no b) Justify your answer (use the space below at the right).

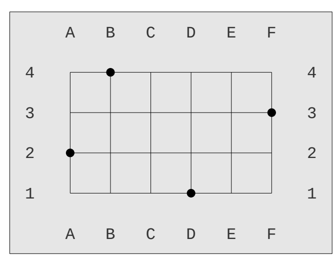
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| S0 | - | - | - | - | - | - | - | * | * |
| S1 | - | - | * | - | - | - | - | - | - |
| S2 | - | * | * | - | - | - | * | - | * |
| S3 | * | - | - | * | - | * | - | * | - |
| S4 | - | - | - | * | - | - | - | - | - |
| S5 | * | - | - | - | - | - | - | - | - |
| S6 | - | - | - | - | * | * | * | - | - |
| S7 | * | - | - | - | - | * | - | - | * |
| S8 | - | - | - | - | - | - | - | - | - |
| S9 | - | - | - | - | * | - | - | - | * |
| S10 | - | * | - | - | - | * | * | - | - |

4. [8 marks] For this 4-pin graph, demonstrate the MST-approx algorithm to find a Steiner tree.

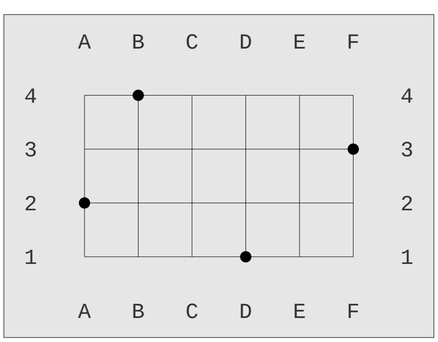
   a) On the left diagram, draw the helper graph $G$ you will use. Label all edges appropriately. With thick edges, show an MST of $G$. b) On the right diagram, draw a Steiner tree corresponding to your MST.

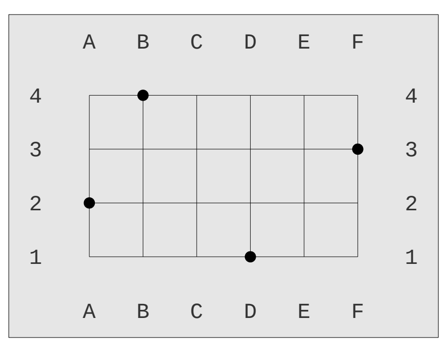
Rough work (optional)




Final answer



1. [5+5 marks]

Here is output from dynamic-program-by-weight (DP) 0-1 knapsack, $n = 5$, $W = 18$. Fill in the missing entries (last two columns, last four rows).

```
val       [6,  5,  7,  6, 11]
wt        [7,  5,  8,  6, 10]
   0   0   0   0   0   0   0
   1   0   0   0   0   0   0
   2   0   0   0   0   0   0
   3   0   0   0   0   0   0
   4   0   0   0   0   0   0
   5   0   0   5   5   5   5
   6   0   0   5   5   6   6
   7   0   6   6   6   6   6
   8   0   6   6   7   7   7
   9   0   6   6   7   7   7
  10   0   6   6   7   7  11
  11   0   6   6   7  11  11
  12   0   6  11  11  11  11
  13   0   6  11  12  12  12
  14   0   6  11  12  13  13
  15   0   6  11  13  __  __
  16   0   6  11  13  __  __
  17   0   6  11  13  __  __
  18   0   6  11  13  __  __
```

Here is python code for DPW. It creates the table T shown at left. Fill in the blanks.

```
def knapDP(val, wt, W):

  n = len(val)

  T = [[0 for j in range(n+1)] for w in range(W+1)]

  for j in range(1,n+1):

    for w in range(W+1):

      T[w][j] = _____________________ if w < wt[j-1] \

        else max(_____________________, ______________________)
```

2. [5 marks] Let $S(n)$ be $\{1, 2, \ldots, n\}$. Let $P(n)$ be the set of all subsets of $S(n)$. Let $f(n)$ be the sum, over all $p$ in $P$, of the size of $p$. E.g. $P(3) = \{\{\}, \{1\}, \{2\}, \{3\}, \{1{,}2\}, \{1{,}3\}, \{2{,}3\}, \{1{,}2{,}3\}\}$, so $f(3) = 0+1+1+1+2+2+2+3 = 12$. **Claim.** $f(n) = n \times 2^{n-1}$. **Proof of Claim.** Assume $n \geq 1$. Consider the $2^n$ subsets of an $n$-set. If we pair each subset $S$ with its complement $T$ (the set of elements not in $S$), we see that there are exactly $2^n/2 = 2^{n-1}$ pairs of subsets. E.g. **give the four pairs of subsets of** $P(3)$:

_________________ _________________ _________________ _________________

So $f(n)$ (the sum of the sizes of the $2^n$ subsets) equals the sum, over all pairs, of the sizes of the two sets in the pair. **Finish the proof of the claim:**

3. [7 marks] *Set cover* is the problem with these instances and query: *Instance.* A set $U$, a set $S = \{S_1, S_2, \ldots, S_t\}$ of subsets of $U$, and an integer $k$. *Query.* Is there a subset $C$ of $S$ of size $k$ whose union equals $U$?

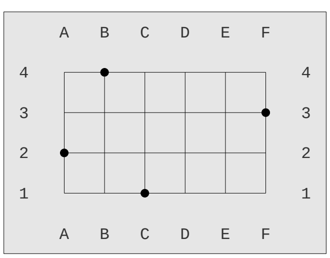
   a) In the set cover instance below, $U = \{0, 1, \ldots, 8\}$ and $S = \{$S0, S1, ..., S10$\}$. Does this instance have a cover of size 6? **(circle one)** yes no b) Justify your answer (use the space below at the right).

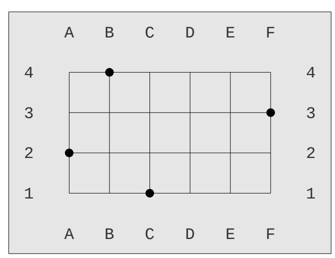
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| S0 | - | - | - | * | - | - | - | - | - |
| S1 | * | - | - | - | - | - | - | - | - |
| S2 | - | - | - | - | * | * | * | - | - |
| S3 | * | - | - | - | - | * | - | - | * |
| S4 | - | - | - | - | - | - | - | - | - |
| S5 | - | - | - | - | * | - | - | - | * |
| S6 | - | * | - | - | - | * | * | - | - |
| S7 | - | - | - | - | - | - | - | * | * |
| S8 | - | - | * | - | - | - | - | - | - |
| S9 | - | * | * | - | - | - | * | - | * |
| S10 | * | - | - | * | - | * | - | * | - |

4. [8 marks] For this 4-pin graph, demonstrate the MST-approx algorithm to find a Steiner tree.

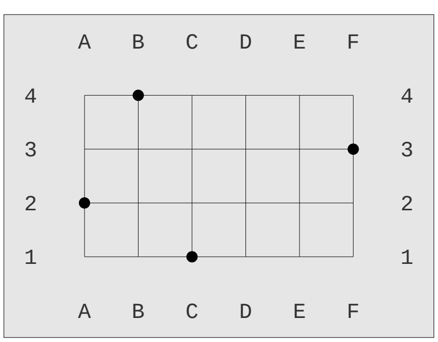
   a) On the left diagram, draw the helper graph $G$ you will use. Label all edges appropriately. With thick edges, show an MST of $G$. b) On the right diagram, draw a Steiner tree corresponding to your MST.

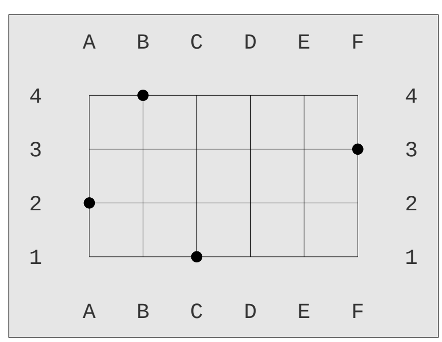
Rough work (optional)




Final answer



**first name** **last name** **id#**

**30 min** **30 marks** **closed book** **no devices** **2 pages** 

1. [5+5 marks]

   Here is output from the dynamic-program-by-weight (DPW) 0-1 knapsack algorithm with $n = 5$, $W = 18$. Fill in the missing entries (last two columns, last four rows).

```
val      [6,  8,  6,  9,  5]
wt       [7,  8,  6, 10,  5]
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 5 |
| 6 | 0 | 0 | 0 | 6 | 6 | 6 |
| 7 | 0 | 6 | 6 | 6 | 6 | 6 |
| 8 | 0 | 6 | 8 | 8 | 8 | 8 |
| 9 | 0 | 6 | 8 | 8 | 8 | 8 |
| 10 | 0 | 6 | 8 | 8 | 9 | 9 |
| 11 | 0 | 6 | 8 | 8 | 9 | 11 |
| 12 | 0 | 6 | 8 | 8 | 9 | 11 |
| 13 | 0 | 6 | 8 | 12 | 12 | 13 |
| 14 | 0 | 6 | 8 | 14 | 14 | 14 |
| 15 | 0 | 6 | 14 | 14 | __ | __ |
| 16 | 0 | 6 | 14 | 14 | __ | __ |
| 17 | 0 | 6 | 14 | 14 | __ | __ |
| 18 | 0 | 6 | 14 | 14 | __ | __ |

   Here is python code for DPW. It creates the data table D shown at left. Fill in the blanks.

```
def knapDP(val, wt, W):

  n = len(val)

  D = [[0 for j in range(n+1)] for w in range(W+1)]

  for j in range(1,n+1):

    for w in range(W+1):

      D[w][j] = _____________________ if w < wt[j-1] \

        else max(_____________________, ______________________)
```

2. [5 marks] Let $S(n)$ be $\{1, 2, \ldots, n\}$. Let $P(n)$ be the set of all subsets of $S(n)$. Let $f(n)$ be the sum, over all $p$ in $P$, of the size of $p$. E.g. $P(3) = \{\{\}, \{1\}, \{2\}, \{3\}, \{1{,}2\}, \{1{,}3\}, \{2{,}3\}, \{1{,}2{,}3\}\}$, so $f(3) = 0+1+1+1+2+2+2+3 = 12$. **Claim.** $f(n) = n \times 2^{n-1}$**. Proof of Claim.** Assume $n \geq 1$. Consider the $2^n$ subsets of an $n$-set. If we pair each subset $S$ with its complement $T$ (the set of elements not in $S$), we see that there are exactly $2^n/2 = 2^{n-1}$ pairs of subsets. E.g. **give the four pairs of subsets of** $P(3)$:

   _________________ _________________ _________________ _________________

   So $f(n)$ (the sum of the sizes of the $2^n$ subsets) equals the sum, over all pairs, of the sizes of the two sets in the pair. **Finish the proof of the claim:**

3. [7 marks] *Set cover* is the problem with these instances and query: *Instance.* A set $U$, a set $S = \{S_1, S_2, \ldots, S_t\}$ of subsets of $U$, and an integer $k$. *Query.* Is there a subset $C$ of $S$ of size $k$ whose union equals $U$?

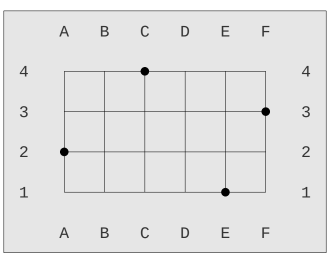
   a) In the set cover instance below, $U = \{0, 1, \ldots, 8\}$ and $S = \{$S0, S1, ..., S10$\}$. Does this instance have a cover of size 6? **(circle one)** yes no b) Justify your answer (use the space below at the right).

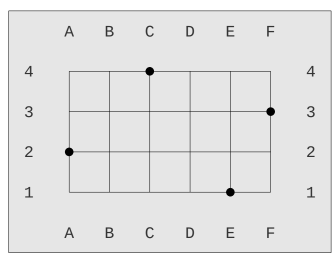
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| S0 | * | - | - | - | - | * | - | - | * |
| S1 | - | - | - | - | - | - | - | - | - |
| S2 | - | - | - | - | * | - | - | - | * |
| S3 | - | * | - | - | - | * | * | - | - |
| S4 | - | - | - | - | - | - | - | * | * |
| S5 | - | - | * | - | - | - | - | - | - |
| S6 | - | * | * | - | - | - | * | - | * |
| S7 | * | - | - | * | - | * | - | * | - |
| S8 | - | - | - | * | - | - | - | - | - |
| S9 | * | - | - | - | - | - | - | - | - |
| S10 | - | - | - | - | * | * | * | - | - |

4. [8 marks] For this 4-pin graph, demonstrate the MST-approx algorithm to find a Steiner tree.

   a) On the left diagram, draw the helper graph $G$ you will use. Label all edges appropriately. With thick edges, show an MST of $G$. b) On the right diagram, draw a Steiner tree corresponding to your MST.

Rough work (optional)




Final answer