Question 1 (Minimum Spanning Tree, 30 points). Find a minimum spanning tree of the following graph:

The solution is given by the highlighted edges above. We can show it is correct by using Kruskal's algorithm. Note that the highlighted edges do form a tree, so we do not need to worry about whether the edges we added formed a cycle or not. The algorithm proceeds as follows:

- Edge 1 is added to $T$.
- Edge 2 is added to $T$.
- Edge 3 is added to $T$.
- Edge 4 is added to $T$.
- Edge 5 is added to $T$.
- Edge 6 is not added as it forms a cycle with edges 2, 1, 5.
- Edge 7 is added to $T$.
- Edge 8 is added to $T$.
- Edge 9 is not added as it forms a cycle with edges 4, 5, 1, 2, 3.
- Edge 10 is added to $T$.
- Edge 11 is added to $T$.
- Edge 12 is not added as it forms a cycle with edges 5, 1, 2, 3.
- Edge 13 is not added as it forms a cycle with edges 7, 10.
- Edge 14 is not added as it forms a cycle with edges 4, 5, 8.
- Edge 15 is added to $T$.
- Edge 16 is not added as it forms a cycle with edges 3, 7.
- Edge 17 is not added as it forms a cycle with edges 15, 4, 5, 1, 2, 7, 11.
Question 2 (Path Counting, 35 points). Find an algorithm that given a graph \( G \) with specified vertices \( s \) and \( t \) and a positive integer \( k \) computes the number of paths from \( s \) to \( t \) in \( G \) containing exactly \( k \) edges. For the purposes of this problem, a path is allowed to repeat vertices. Your runtime should be \( O(k(|V| + |E|)) \) or better for full credit.

For each \( m \leq k \) and each \( v \) in \( V \), we let \( N_m(v) \) be the number of \( s - v \) paths with exactly \( m \) edges. We note that

\[
N_0(v) = \begin{cases} 
1 & \text{if } v = s \\
0 & \text{otherwise}
\end{cases}
\]

and that for \( m > 0 \), \( N_m(v) \) is given by the recurrence

\[
N_m(v) = \sum_{(u,v) \in E} N_{m-1}(u).
\]

This is because a path of length \( m \) ending at \( v \) is given by a path of length \( m - 1 \) ending at some \( u \) with an edge to \( v \) followed by the edge \((u, v)\). Thus, we have the following dynamic program:

Initialize an array \( N \) indexed by a number \( m \) at most \( k \) and a vertex of \( G \)
For \( v \) in \( V \):
    \( N[0,v] \leftarrow 0 \)
\( N[0,s] \leftarrow 1 \)
For \( m = 1..k \)
    For \( v \) in \( V \)
        \( \text{tot} \leftarrow 0 \)
        For \((u,v)\) in \( E \)
            \( \text{tot} \leftarrow \text{tot} + N[m-1,u] \)
        \( N[m,v] \leftarrow \text{tot} \)
\( \text{return } N[t,k] \)

To analyze the runtime, not that for each \( m \) from 1 to \( k \) we need to analyze each vertex \( v \), and need to sum over each edge going into \( v \). Therefore, the runtime is \( O(k(|V| + |E|)) \).
Question 3 (Convex Approximation, 35 points). Call a function $f : \{1, 2, \ldots, n\} \rightarrow \mathbb{Z}$ convex if $f(k - 1) + f(k + 1) \geq 2f(k)$ for all $1 < k < n$. Give an algorithm that given positive integers $n$ and $m$ and a function $g : \{1, 2, \ldots, n\} \rightarrow \mathbb{Z}$, computes the minimum possible value of $E = \sum_{i=1}^{n} |f(i) - g(i)|$ over all convex functions $f : \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, m\}$.

For example, if $n = m = 3$ and $g$ is given by $g(1) = 1, g(2) = g(3) = 2$, the answer should be 1, which is attained for example by the function $f$ given by $f(1) = 1, f(2) = 2, f(3) = 3$.

Your algorithm should run in time polynomial in $n$ and $m$ for full credit.

For each $k \leq n$ and $1 \leq a, b \leq m$, we let $E[k, a, b]$ be the minimum over convex functions $f \{1, 2, \ldots, k\} \rightarrow \{1, 2, \ldots, m\}$ with $f(k) = a, f(k - 1) = b$ of $\sum_{i=1}^{k} |f(i) - g(i)|$. We note that this can be computed recursively. This is because a convex function $f$ on $\{1, 2, \ldots, k+1\}$ is given by a convex function on $\{1, 2, \ldots, k\}$ in addition to a value at $k + 1$ so that $f(k + 1) + f(k - 1) \geq 2f(k)$. Therefore, for $k \geq 2$, we have the recurrence

$$E[k + 1, a, b] = |a - g(k + 1)| + \min_{1 \leq c \leq m, c + a \geq 2b} E[k, b, c].$$

This gives us the following algorithm

Initialize an $n \times m \times m$ array $E$

For $a = 1 \ldots m$
    For $b = 1 \ldots m$
        $E[2, a, b] \leftarrow |g(2) - a| + |g(1) - b|$

For $k = 2 \ldots n-1$
    For $a = 1 \ldots m$
        For $b = 1 \ldots m$
            $Best = \infty$
            For $c = 1 \ldots m$
                if $a + c \geq 2b$:
                    $Best \leftarrow \min(Best, E[k, b, c])$
                    $E[k+1, a, b] \leftarrow Best + |a - g(k+1)|$
            $Best \leftarrow \infty$
            For $a = 1 \ldots m$
                For $b = 1 \ldots m$
                    $Best = \min(Best, E[n, a, b])$
        return $Best$

The runtime of this algorithm is determined by the loop structure. The most complicated nested loop is $n$ by $m$ by $m$, so the runtime is $O(nm^3)$. 

3