A

首先考虑离线，那么一个点u能到达v，意味着在u到v路径上存在时间递增的2操作序列。

接着我们考虑点分治，将分治中心作为根。我们先求出每个点到根的最早时间，这个可以用一个简单的dp完成，也就是从上往下，求出沿着每个2操作传上去什么时候能到达根，转移的时候只要找父亲节点下一个2操作。

我们将所有点按传递到根的时间排序。按到达根的时间从早到晚做一个类似于扫描线的东西，同时用维护一下每个点的到达时间，记一下前缀和。随着到达根的时间推迟，每个点的到达时间也会推迟，但总的变化量不会超过$O(m)$。维护的时候可以用一个类似于dfs的方法维护，也就是一个点的时间如果变晚了，看一下它传递下去的时间如果比当前时间还要早，那么就要推迟。

总的时间复杂度为$O((n+m)\log^2 n)$。

B

新的图是G与一条路径P_n的笛卡尔积。令λ_i为G的Laplacian矩阵的特征值，那么一个图的生成树为$\frac{1}{n}\prod_{i=1}^{n-1}\lambda_i$。

引理：对于两个图G_1和G_2的笛卡尔积G，它的Laplacian矩阵的特征值为$\lambda_i+\mu_j$其中λ_i与μ_j取遍G_1和G_2的特征值。所以令$L(G,x)$为G的Laplacian矩阵的特征多项式，那么这个结果可以看成$L(G_1,x)$与$L(G_2,-x)$的resultant。

令$P_n(x)=L(P_n,x),Q(x)=L(G,x)$，根据resultant的性质有$\operatorname{res}(P_n,Q)=\operatorname{res}(P_n \bmod Q,Q)$，所以只要求出$P_n \bmod Q$即可，通过~~找规律~~计算可以发现$P_n=(x+2)P_{n-1}-P_{n-2}$，所以可以用线性递推的方法在$O(|Q|\log n\log|Q|)$时间复杂度内算出来，当然如果使用暴力多项式出发复杂度为$O(|Q|^2\log n)$。

注意BM算法只能算矩阵的最小多项式，无法算矩阵特征多项式，所以你得~~上google~~抄一个$O(n^3)$的特征多项式方法。然后算resultant的时间复杂度为$O(|Q|^2)$，所以就做完了。

C

令$L=\operatorname{lcm}(1,2,\ldots,k+1)$，我们可以发现如果$x \bmod L=r$一定的情况下，一定是一个多项式，或者说$\frac{x(x+1)\ldots(x+k)}{\operatorname{lcm}(x,x+1,\ldots,x+k)}$是一个定值$c$，那么可以写成$\sum\binom{L\cdot i+r}{k+1}$的形式，可以通过拆一下项在$O(Lk)$时间复杂度内算出来。

同时，我们注意到，每个素数对定值c的贡献是独立的，也就是说$x \bmod p^e$固定的条件下，除掉的数一定是某个定值。

所以我们可以用meet in the middle + CRT来解决啦。根据CRT，我们有如果$x \bmod L_1=r_1,x \bmod L_2=r_2$，那么$x\equiv r_1M_1+r_2M_2 \pmod{L_1L_2}$。所以可以按照$r_1M_1 \bmod L_1L_2$和$r_2M_2 \bmod L_1L_2$排序，然后暴力把那个式子拆开来，做到$O(k^2)$合并即可。

虽然有一些进一步的优化，但是上面的做法卡卡常数就可以过了。