Řešení úlohy č. 2
Routování

Úloha se skládala ze dvou částí, nejdříve bylo potřeba najít největší masku, která rozděluje síť na maximálně k podsítí. To nám vytvořilo graf. Následně bylo třeba spočítat pro každou hranu grafu, o kolik je nejkratší cesta kratší.

1 Největší maska

Uvědomme si, že jakákoliv 32bitová maska začínající jedničkou je větší než kterákoliv, která začíná nulou. Pokud tedy existuje maska začínající jedničkou, která rozděluje síť na maximálně k podsítí, už nás nemusí zajímat ty, které začínají nulou. Stačí nám tedy spočítat, kolik unikátních podsítí nám vznikne aplikováním bitového součinu (operátor &) s maskou `10000000 00000000 00000000 00000000` na každou adresu. Pokud je výsledek menší nebo roven k, první bit je nutně 1, v ostatních případech 0. Dále pokračujeme obdobně, známe první bit a zkoušíme jedničku na druhé pozici. Celkem 32krát nám proběhne cyklus, který spočítá, zda může být na dané pozici jednička. Zjištění počtu unikátních podsítí můžeme implementovat např. pomocí řazení v čase $O(n \log(n))$ (seřadíme a zjišťujeme, kolikrát máme za sebou dvě různé podsítě). Pokud použijeme datovou strukturu množina s vkládáním v čase až $O(1)$, můžeme dokonce časovou složitost zmenšit až na $O(n)$, ale varianta s $O(n \log(n))$ na vyřešení této úlohy bohatě stačila.

2 Součet zkratek

S výslednou maskou zjistíme bitovým součinem bitové řetězce charakterizující podsítě, unikátních je nejvýše k. Pro každou dvojici můžeme podle zadání spočítat vzdálenost (počítáme pouze s bity, kterým odpovídá v masce jednička). Vytvořili jsme si vážený úplný graf s až k vrcholy a hledáme součet délek nejkratších cest mezi každými dvěma vrcholy, od kterých odečteme váhu všech hran. Pro nalezení všech nejkratších cest z jednoho vrcholu lze použít Dijkstrův algoritmus. Ten běží v čase $O(|E| + |V| \log |V|)$[1], tedy pro nás $O(k^2)$. Abychom zjistili pro každou dvojici vrcholů nejkratší cestu mezi nimi, potřebujeme spustit Dijkstrův algoritmus za každý vrchol jednou a celkový čas je tedy $O(k^3)$. Existuje však jednodušší řešení, a to Floyd-Warshallův algoritmus, který přímo běží v čase $O(k^3)$ a není potřeba žádných netriviálních datových struktur. Více o těchto algoritmech lze nalézt v doporučené literatuře: `https://pruvodce.ucw.cz/static/pruvodce.pdf#page=143`.

[1] Pro získání tohoto času je potřeba použít netriviální datová struktura – halda, kde dokážeme zmenšit hodnotu klíče v průměru za konstantu.