

KSP-35-3-3

Richard Tichý

1 Popis algoritmu

Pro řešení tohoto stromového problému, kde se snažíme přepojit zařízení (listy) z prodlužovaček (uzlů) do zásuvek tak, abychom pro každou prodlužku dodrželi její maximální příkon, a abychom minimalizovali počet přepojovaných zařízení, jsem se rozhodl použít depth-first search.

Pro každý vrchol, resp. prodlužku, uděláme sestupně seřazený dle příkonu seznam všech koncových zařízení, která se nachází v jejích podstromech (dále jen „seznam koncových zařízení“). Tento seznam vytvoříme ze seznamů z podstromů (DFS) a následným slitím¹ dohromady. Dále si sečteme příkon všech podstromů (DFS). Pokud je součet příkonů podstromů vyšší než nejvyšší snesitelný (dále „maximální“) pro tento uzel, budeme odebírat, resp. přepojovat do zásuvek, zařízení dle seznamu koncových zařízení, tedy od těch s nejvyšším příkonem, dokud příkon v daném podstromě nebude roven nebo nižší než maximální pro tento uzel. Jakmile příkony pro tento uzel sedí, půjdeme zpět na předka a provedeme pro něj to samé.

Minimální nutný počet zařízení, která je potřeba odpojit, si spočítáme po DFS spočítáním počtu zařízení v seznamu těch k přepojení.

1.1 Důkaz správnosti

Nejprve si dokažme, že pro jakýkoli vstup tento algoritmus vrátí řešení, které bude mít ve všech prodlužkách celkový příkon menší nebo roven nejvyššímu snesitelnému: jelikož u každé prodlužky sledujeme součet příkonů zařízení na ni napojených, a přepojujeme zařízení, která na ni jsou připojená, dokud je tento součet větší než nejvyšší snesitelný, musí být pro všechny prodlužky splněna podmínka snesitelnosti příkonu.

Další možnou nesprávností algoritmu může ještě býti řešení, které vyžaduje přepojení více zařízení, než je nutné. Tuto nesprávnost vyvrátíme důkazem sporem: předpokládejme, že existuje vstup, pro který jeho řešení vyžaduje méně přepojení, než vyžaduje řešení od výše popsaného algoritmu. Správné řešení by muselo přepojit podobný příkon za nižší počet přepojení, což už kvůli postupnému přepojování zařízení s nejvyšším příkonem nemůže být; a jelikož nejprve splňujeme koncové podmínky (snesitelnost příkonu nejlouběji ve stromě), kde se to vždy musí splnit, odebereme nejprve ty, které bychom museli i později, dochází ke sporu.

1.2 Časová a prostorová složitost

Označme počet prodlužek jako „v“ a počet zařízení jako „z“.

1.2.1 Časová složitost

Jelikož používáme DFS, které projde každou prodlužkou jednou, a u každé prodlužky během DFS průchodu sléváme seznamy koncových zařízení, časová složitost bude $O(v \cdot z)$.

1 Slití seznamů, přejaté z mergesortu - <https://ksp.mff.cuni.cz/kucharky/trideni/>

1.2.2 Prostorová složitost

Potřebujeme si pamatovat všechny prodlužky a zařízení, z čeho vyplývá, že prostorová složitost bude $O(v+z)$.