# Srovnání účtů

Při zpracování tohoto příkladu jsem počítal s následujícím předpokladem:

Pokud mám An účtů, ze kterých přesouvám peníze do Bn účtů, tak nejlevnějším způsobem jak přesunout peníze je začít s účtem A, který má nejdražší cenu za odchozí platbu a účtem B, který má nejlevnější cenu za příchozí platbu, a poté postupně brát účty A s levnějšími odchozími cenami a účty B s dražšími příchozími cenami.

# Program

Nejprve sestavím dva arraye, jeden pro odchozí platby a jeden pro příchozí platby. Každá položka v arrayjích má 2 hodnoty: cenu za odchozí / příchozí platbu (podle toho o který array se jedná) a počet peněz na účtě. Oba dva arraye jsou seřazeny podle první hodnoty (cena za odchozí / příchozí platbu) a oba mají délku n, kde n je počet účtů.

Nyní se snažím najít minimální možný zůstatek na účtech (což má být výsledek), vím že není menší než zůstatek na účtu s nejmenším zůstatkem a není větší než zůstatek na účtu s největším zůstatkem. Mezi těmito dvěma hodnotami tedy provádím binární vyhledávání a snažím se najít první možný nejmenší zůstatek.

To, jestli je zůstatek možný nebo ne, zjistím pomocí mých dvou seřazených arrayů. Nejdříve od konce procházím array seřazený podle odchozích cen, abych získal účet s nějvětší možnou odchozí cenou, který ale zároveň má zůstatek větší, než momentálně zkoumaný zůstatek. Poté od začátku procházím array seřazený podle příchozích cen, abych získal účet s nejmenší možnou příchozí cenou, který má zároveň menší zůstatek, než momentálně zkoumaný zůstatek.

Poté virtuálně provedu převod tím, že odečtu celkovou cenu převodu z poskytlých peněz. Pokud mi v účtu, do kterého převádím peníze, zbyde volné místo, tak v další iteraci vezmu další účet ze kterého musím převést peníze. Pokud mi naopak zbydou peníze na převedení, tak v další iteraci vezmu další účet s volným místem. Pokud účet naplním přesně a nezbyde mi žádné místo navíc, tak vezmu nové účty z obou dvou arrayů.

Tento postup opakuji dokud mi nedojdou peníze pro převádění, nebo dokud úspěšně nepřevedu všechny přebylé peníze, nebo dokud mi nedojdou účty, do kterých můžu převádět peníze.

Podle toho, která z těchto tří možností nastane, se roznodne, zde je možné mít takovýto maximální zůstatek účtech. Poté dalším krokem binárního vyhledávání získám nový možný zůstatek a process opakuji, dokud nezískám nejmenší možný zůstatek.

Algoritmus má časovou obtížnost O = n\*log(h), pokud n je počet účtů a h je rozdíl mezi největším a nejmenším zůstatkem z řady účtů. K řešení jsem použil python knihovnu numpy.