

# Určení ceny

## Kritéria

1. Maximalizace zobchodovaného finančního objemu ( $\text{Množství} * \text{Cena}$ )
2. Minimalizace nezobchodovaného finančního objemu ( $\text{Množství} * \text{Cena}$ )
3. Převaha nabídky / poptávky
4. Průměrná cena z možných
5. Zaokrouhlení podle předchozí ceny

## Vstupy

- (Seřazený?) seznam objednávek na nákup (buyList)
- (Seřazený?) seznam objednávek na prodej (sellList)
- Předchozí cena (lastPrice)
- Cenová jednotka (tickSize)

## Algoritmus

### ***Varianta 1: seřazené seznamy (implementováno)***

#### **První krok: inicializace**

Seznamy objednávek nemusí obsahovat všechny došlé objednávky. Pro výpočet ceny (a případně i dále pro párování) mají smysl jen objednávky na nákup, jejichž poptávaná cena je vyšší nebo rovna nejnižší nabízené ceně objednávky na prodej. Stejně tak na druhé straně mají smysl jen objednávky na prodej, jejichž nabízená cena je nižší nebo rovna nejvyšší poptávané ceně objednávky na nákup. **K zamyšlení:** Tuto filtraci lze provést rovnou vhodným dotazem na databázi nebo až v rámci výpočtu. (*Přikláním se pro filtraci v databázi.*)

buyList seřadíme **sestupně** podle ceny. sellList seřadíme **vzestupně** podle ceny. **K zamyšlení:** Řadit zde, nebo si říct databázi rovnou o seřazené seznamy? (*Přikláním se pro řazení v databázi.*)

Vytvoříme tabulku (pole, seznam), ve které budeme evidovat tyto údaje: Cena (price), Koupitelné množství (aggregateBuyQuantity), Prodatelné množství (aggregateSellQuantity), případně další pomocné údaje.

#### **Druhý krok: vložení objednávek na nákup**

Procházíme seznam objednávek na nákup (buyList) takto:

Pokud se daná cena v tabulce ještě nevyskytuje (stačí kontrolovat poslední řádek tabulky), přidáme nový řádek. Cena v novém řádku tabulky je rovna ceně v objednávce. Koupitelné množství je rovno součtu množství v objednávce a koupitelného množství v předchozím řádku tabulky. Prodatelné množství je zatím nulové.

Jestliže se již daná cena v tabulce vyskytuje (v tom případě je na posledním řádku), zvýšíme koupitelné množství v tomto řádku o množství v objednávce.

Poznámka: Ceny v tabulce jsou (na konci 2. kroku) stejně jako buyList seřazeny sestupně.

#### **Třetí krok: vložení objednávek na prodej**

Procházíme seznam objednávek na prodej (sellList) dopředně a tabulku z předešlého kroku **od konce** takto:

Pokud je cena na aktuálním řádku tabulky menší než cena v aktuální objednávce, vyplníme prodatelné množství v tomto řádku tabulky stejnou hodnotou, jako je o řádek níže (resp. 0, pokud začínáme a toto je poslední řádek), a posuneme se o řádek výš. Pro něj znova vyhodnotíme tuto podmínku.

Pokud je cena na aktuálním řádku tabulky větší než cena v aktuální objednávce (což znamená, že se daná cena v tabulce nevyskytuje), vložíme do tabulky nový řádek na odpovídající pozici (pod aktuální řádek). Cena na novém řádku je stejná jako cena v objednávce, koupitelné množství je stejné jako v aktuálním řádku (tj. nejbližší vyšší ceně) a prodatelné množství je rovné součtu množství v objednávce a v předchozím (tj. nižším, s menší cenou) řádku tabulky. V tabulce se posuneme na nově vložený řádek.

Zbývá případ, že jsme v tabulce našli řádek s cenou rovnou aktuální objednávce. V takovém případě zvýšíme prodatelné množství v aktuálním řádku o množství v objednávce.

### Čtvrtý krok: určení ceny

Určení ceny z tabulky je již triviální, stačí se řídit zvolenými kritérii. V případě neúspěchu 1. kritéria postupují do dalších kol porovnání pouze nejvhodnější kandidáti.

### ***Varianta 2: předvýpočet cenových hladin***

Operaci vkládání nových cenových hladin při procházení poptávek je možné obejít tak, že si nejprve projdeme všechny nabídky i poptávky a vytvoříme si seřazené pole cenových hladin v nich obsažených (respektive si o tyto zastoupené použitelné hladiny řekneme databázi). Toto pole vložíme do tabulky a dále již postupujeme podobně jako v předchozím případě, avšak bez nutnosti vkládat nové řádky.

Něco jako:

```
SELECT MAX(price) AS maxprice FROM orders WHERE oid = 1; -- buy
SELECT MIN(price) AS minprice FROM orders WHERE oid = 2; -- sell
SELECT DISTINCT price FROM orders WHERE price <= maxprice AND price >=
minprice ORDER BY price;
```

### Výstupy

- Nová cena (price)
- Tabulka s koupitelným a prodaletným množstvím pro jednotlivé ceny
- Seřazený a filtrovaný seznam objednávek na nákup (buyList)
- Seřazený a filtrovaný seznam objednávek na prodej (sellList)

### Komentáře

# Spárování objednávek

## Kritéria

1. Cena (od nejvyšší u nákupu, od nejnižší u prodeje)
2. Čas přijetí (odeslání?) požadavku
3. (Co když budou nastejno?)

## Vstupy

- (Seřazený a filtrovaný?) seznam objednávek na nákup (buyList)
- (Seřazený a filtrovaný?) seznam objednávek na prodej (sellList)
- Cena (price)

## Algoritmus

Bud' dostaneme již použitelné výběry seznamů seřazené podle kritérií, nebo to provedeme na začátku párovacího procesu. Pak se dá využít toho, že (alespoň) jeden ze seznamů objednávek bude vyčerpán zcela.

### ***Varianta 1: párování podle priority (implementováno)***

Procházíme a párujeme seznamy objednávek tak, jak jsou seřazené podle priority. Přebývajících množství v jedné objednávce bude spárováno s následující. Párování končí vyčerpáním (alespoň) jednoho ze seznamů.

Všechny spárované objednávky budou mít zbývajících množství rovno 0, pouze v poslední párované objednávce z nevyčerpaného seznamu může zbývat nenulové množství.

Pokud je počet objednávek k nákupu  $b$  a k prodeji  $s$ , pak maximální počet vytvořených párů je  $b + s - 1$ .

### ***Varianta 2: párování stejných objemů***

V prvním kroku je potřeba vybrat objednávky, které budou vyřízeny. Bude to vždy celý jeden seznam (nákup nebo prodej), z druhého je třeba odpočítat vhodné množství. Pokud nemusíme říkat databázi, kolik z objednávky zbývá, pak můžeme nevyužité objednávky zahodit a objednávku, jež bude spárována jen částečně, snížit na množství, které bude spárováno. V takový okamžik se zabýváme jen objednávkami, které budou v obou seznamech vyřízeny úplně.

Seznamy objednávek seřadíme podle množství (jedno, zda vzestupně, nebo sestupně, stačí, aby byly seřazené stejně). V první iteraci je procházíme a párujeme ty, které mají stejné množství. Zbývajících párujeme stejným způsobem jako v předchozí variantě.

Pokud je počet objednávek k nákupu  $b$  a k prodeji  $s$ , pak maximální počet vytvořených párů je stejně jako v předchozí variantě  $b + s - 1$ , ale v případě reálných dat by měl být nižší. Na druhou stranu je výpočet náročnější o nutnost řazení a dalšího procházení.

## Výstupy

- Tabulka „kdo komu kolik“ s cenovkou v hlavičce
- Provedené změny v databázi

## Komentáře

Při výpočtu ceny se nám hodilo seřadit objednávky podle ceny; pro účely párování je potřebujem navíc seřadit podle sekundárního kritéria – času. Nabízí se možnost vytáhnout si z databáze objednávky seřazené podle obou kritérií již na začátku při určování ceny. Naproti tomu ale pro párování používáme jen podmnožinu objednávek použitých pro určení ceny (nákupy za ceny  $\geq$  spočítaná cena, prodeje za ceny  $\leq$  spočítaná cena), takže při ručním řazení této podmnožiny budeme řadit menší množství dat.

Zápis do databáze – kdo ho bude provádět a jak bude probíhat? Navrhují, ať je to kalkulátor ve fázi, kdy posílá marketu tabulku prodejů. Databázi se předají uzavřené obchody spočítané v této fázi, ty se zapíší a podle nich se upraví uložené aktivní objednávky (zmenší se objednané množství, případně se odstraní). V takovém případě se informace o zbývajícím množství v objednávkách získaná při párování dále nevyužije.