

Název projektu: Skyline

Realizoval: Krivonosenko Daniil

### 1. Popis projektu:

Projekt je webová aplikace pro vyhledávání hotelů na základě parametrů určených uživatelem. Cílem projektu je vytvořit algoritmus pro efektivní výpočet a vizualizaci výsledků vyhledávání. Vstupem projektu je seznam hotelů s různými parametry, na jejichž základě se provádí řazení. Výsledkem projektu je seznam hotelů, které splňují zadaná kritéria a jsou seřazeny podle kvality svých parametrů, výsledkem je také graf s vyznačenou Skyline. Data možná řadit a vyhledávat Skyline od největšího k nejmenšímu nebo naopak, tato nastavení lze nastavit nezávisle pro každou osu.

### 2. Způsob řešení:

V rámci projektu byly použity následující přístupy a algoritmy pro řešení úkolu:

- Brute-force algoritmus: Pro porovnání výhledů všech hotelů se všemi ostatními hotely a nalezení jejich kvality vzhledem k ostatním.
- Divide-and-conquer algoritmus: Pro rozdělení množiny hotelů na menší části a efektivní výpočet kvality výhledů v rámci jednotlivých částí.
- BNL (Block Nested Loop) algoritmus: Pro efektivní vyhledávání nejlepších hotelů na základě dominance.
- Plane sweep algoritmus: Pracuje na principu procházení horizontální roviny odshora dolů. Po seřazení podle osy x.

### 3. Implementace:

Projekt byl implementován v programovacím jazyku Java script s použitím Node js pro back-end a React pro front-end. Tyto technologie jsem použil proto, aby bylo možné do aplikace snadno přidávat nové vyhledávací algoritmy, nové sekce a komponenty. Aplikace je navržena jako webová aplikace s jednoduchým uživatelským rozhraním. Pro běh aplikace jsou vyžadovány data hotelů ve formě CSV souboru nebo generované z pomoci funkce.

## Choose right hotel for your vacation

Select first parameter  ☐ Low to high Select second parameter  ☐ Low to high Select algorithm

Previous 1 2 3 4 5 6 7 8 Next

id	hotel_name	location	distance_to_beach	distance_to_center	price	stars	rating
1	Herman, Ortiz and Simonis	6 3rd Court	4.5	135.2	1735	5	1
2	Kessler-Lakin	8698 Loomis Circle	115.4	24.5	724	5	2
3	Reilly, Lakin and Hagenes	41943 School Terrace	89.5	51.6	1616	5	6
4	Schamberger-Nitzsche	38 Thierer Park	43.2	76.9	374	4	5

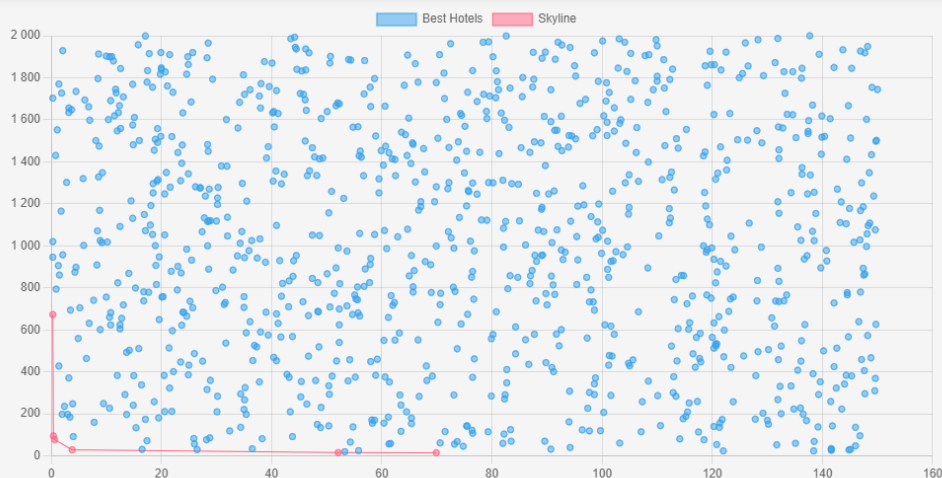
Základní rozhraní aplikace

#### 4. Příklad výstupu:

Příklad výstupu aplikace – seznam hotelů seřazený podle vybraných parametrů a graf s červenou Skylline. Výstup je prezentován v textové podobě a také ve formě grafické vizualizace, která zobrazuje umístění všech hotelů na grafu s skyline. Níže jsou uvedeny obrázky s ukázkou výsledků.

## Choose right hotel for your vacation

Select first parameter  ☒ High to low Select second parameter  ☒ High to low Select algorithm



## 5. Experimentální sekce:

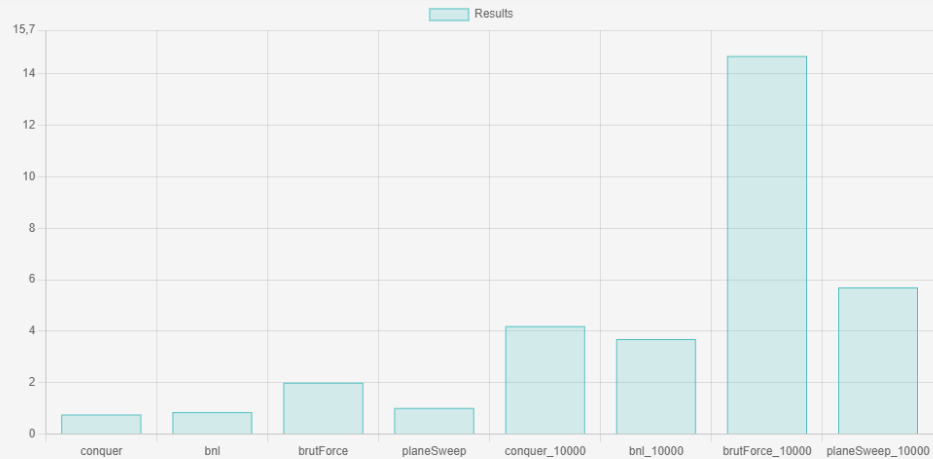
Při testování byly zkoumány různé vstupní parametry projektu, jako například velikost vstupního datasetu hotelů, hodnoty kritérií pro vyhledávání hotelů a efektivita použitých algoritmů v závislosti na velikosti vstupu. Bylo provedeno porovnání různých algoritmů pro výpočet kvality výhledů a jejich efektivitu v různých situacích. Také bylo zkoumáno chování aplikace v situacích s neplatnými nebo neúplnými vstupy. Pro testování rychlosti algoritmů a vizualizaci výsledků jsem v aplikaci připravil speciální funkcionalitu. Níže jsou uvedeny testy v různých situacích.



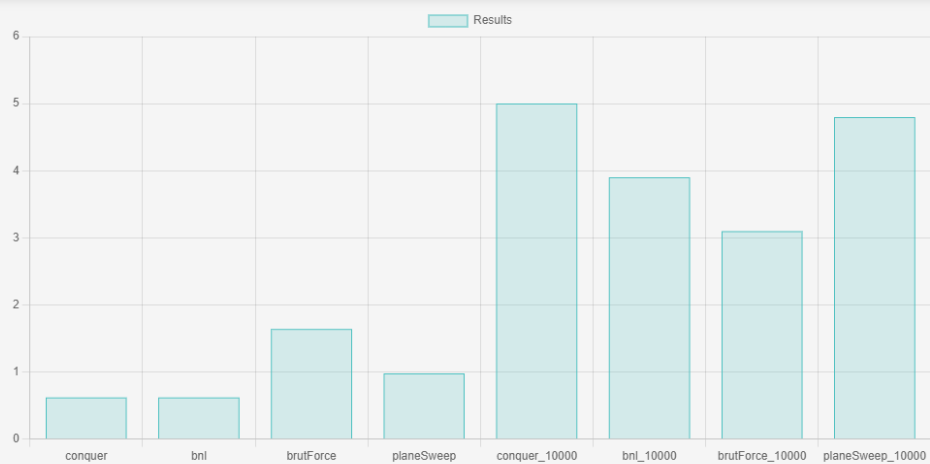
Obrázky testování algoritmů.

## Výsledky porovnání rychlostí nekorelovaných dat z různé délky Skyline.

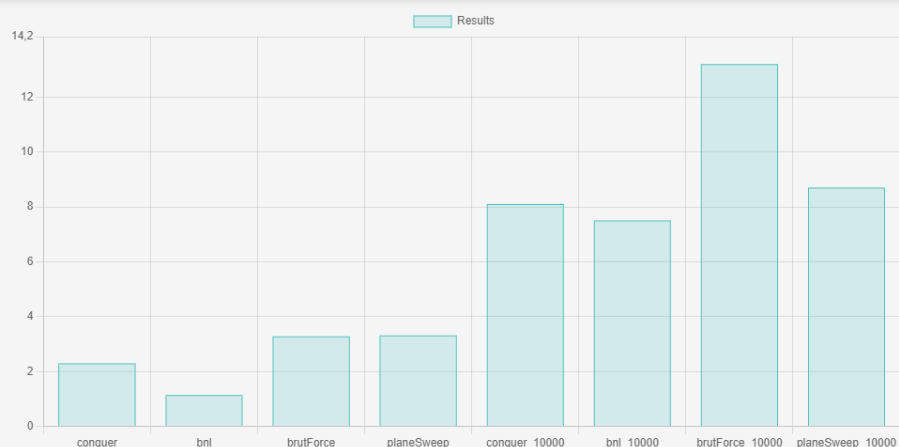
Select first parameter ☒ Low to high  Select second parameter ☒ Low to high  Select algorithm



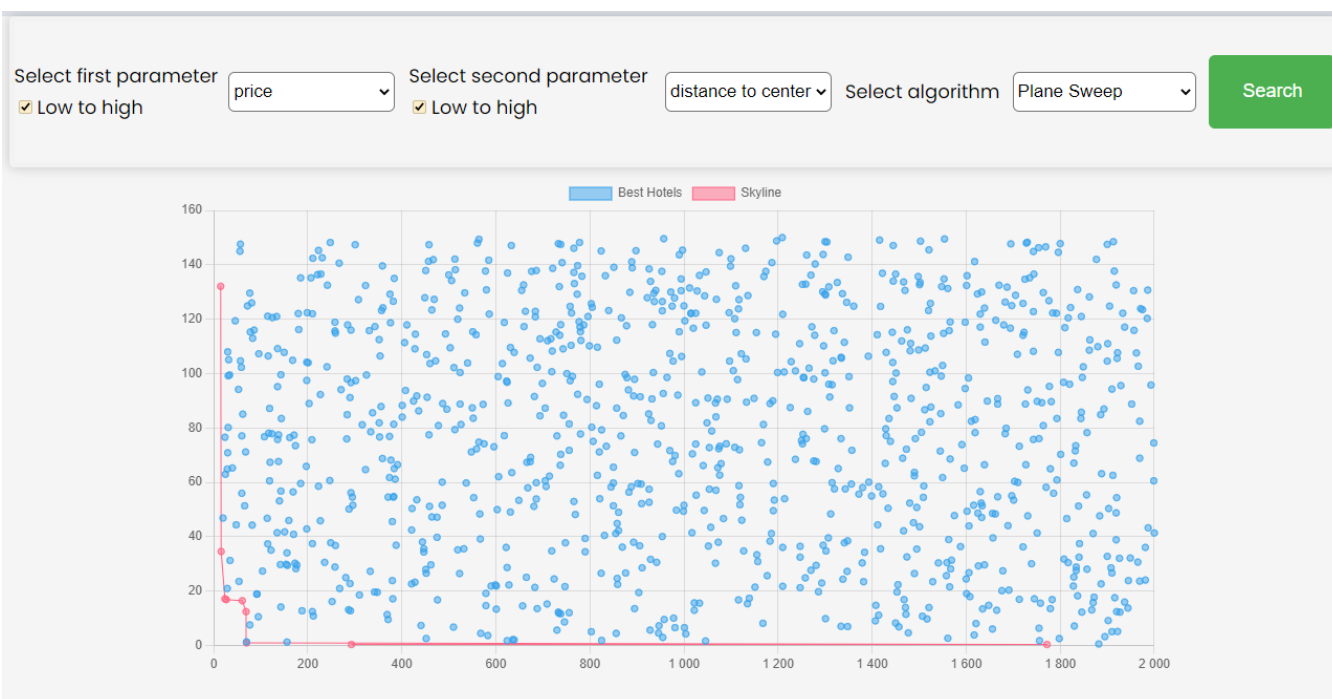
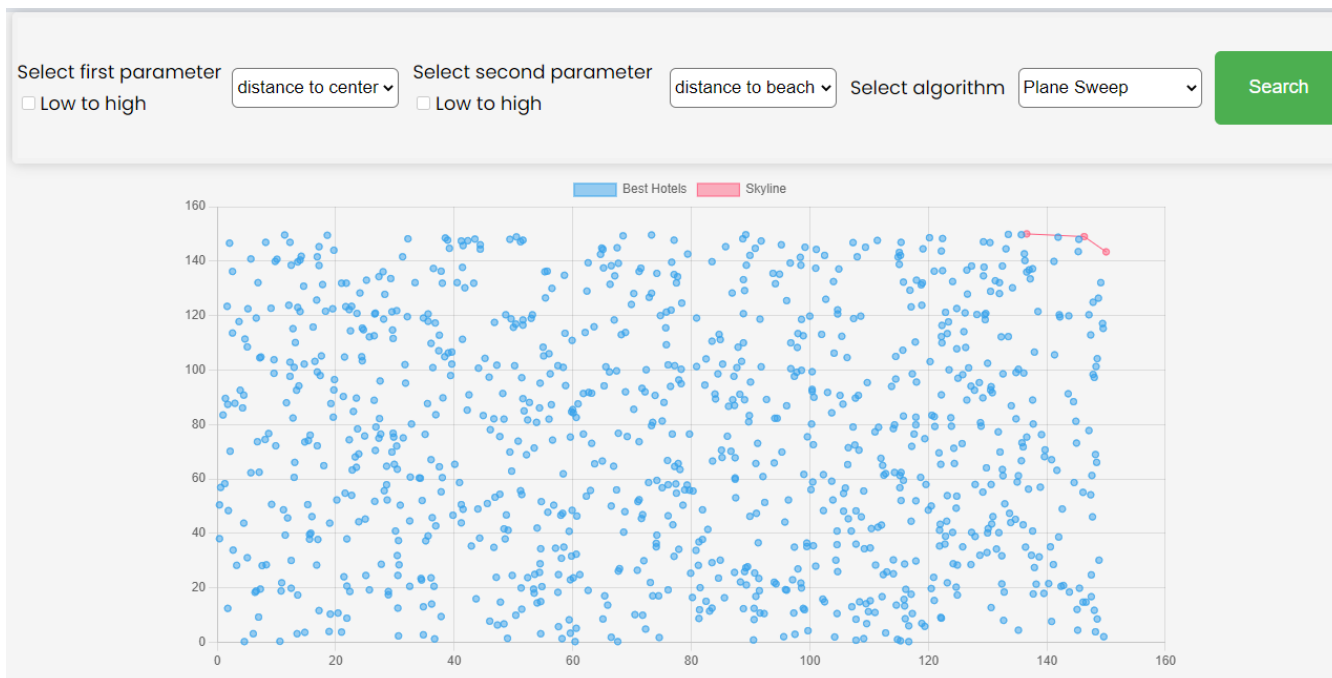
Select first parameter ☒ Low to high  Select second parameter ☒ Low to high  Select algorithm



Select first parameter ☐ Low to high  Select second parameter ☐ Low to high  Select algorithm



V průměru nejhůře si v různých testech vedl algoritmus brute force, a to jak u malých objemů dat (1000), tak u velkých objemů dat (10000). Následují dva algoritmy jsou divide and conquer and plane sweep. BNL algoritmus se nejlépe projevil na velkém množství dat. Obrázky s řádkem pro tyto testy.



Obrázky se skyline pro tyto testy.

Nyní budou uvedeny testy pro korelovaná data.

## Choose right hotel for your vacation

Select first parameter

☒ Low to high

stars

Select second parameter

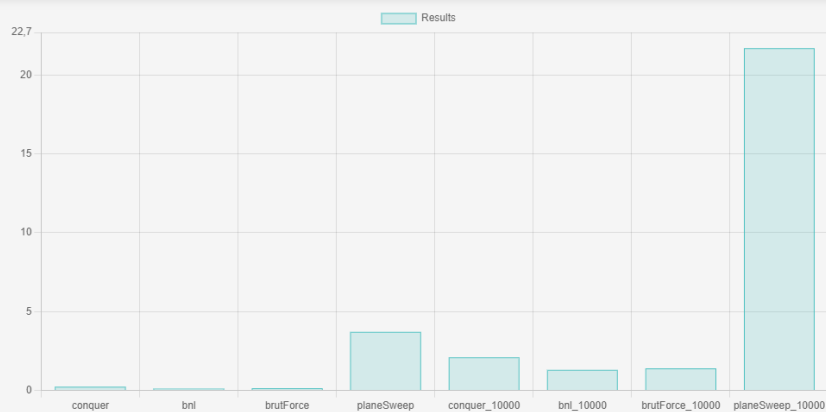
☒ Low to high

price

Select algorithm

Test all

Search



Select first parameter

☐ Low to high

price

Select second parameter

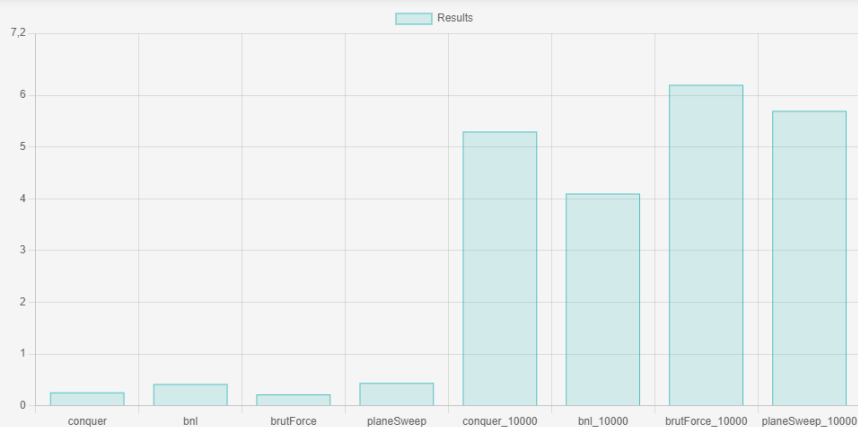
☐ Low to high

stars

Select algorithm

Test all

Search



Select first parameter

☒ Low to high

price

Select second parameter

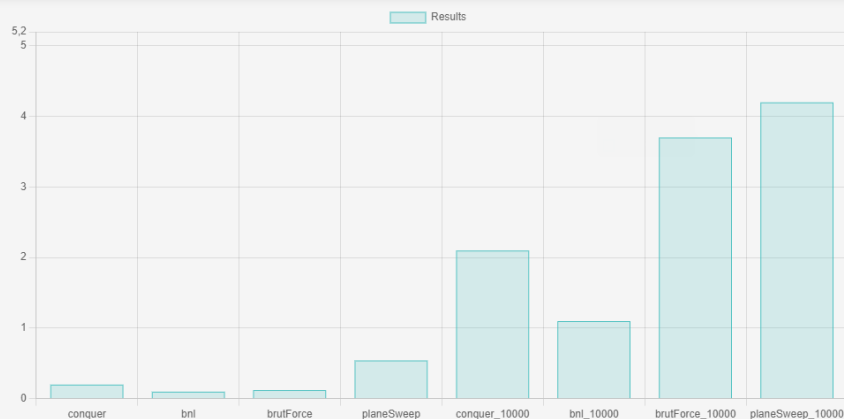
☒ Low to high

stars

Select algorithm

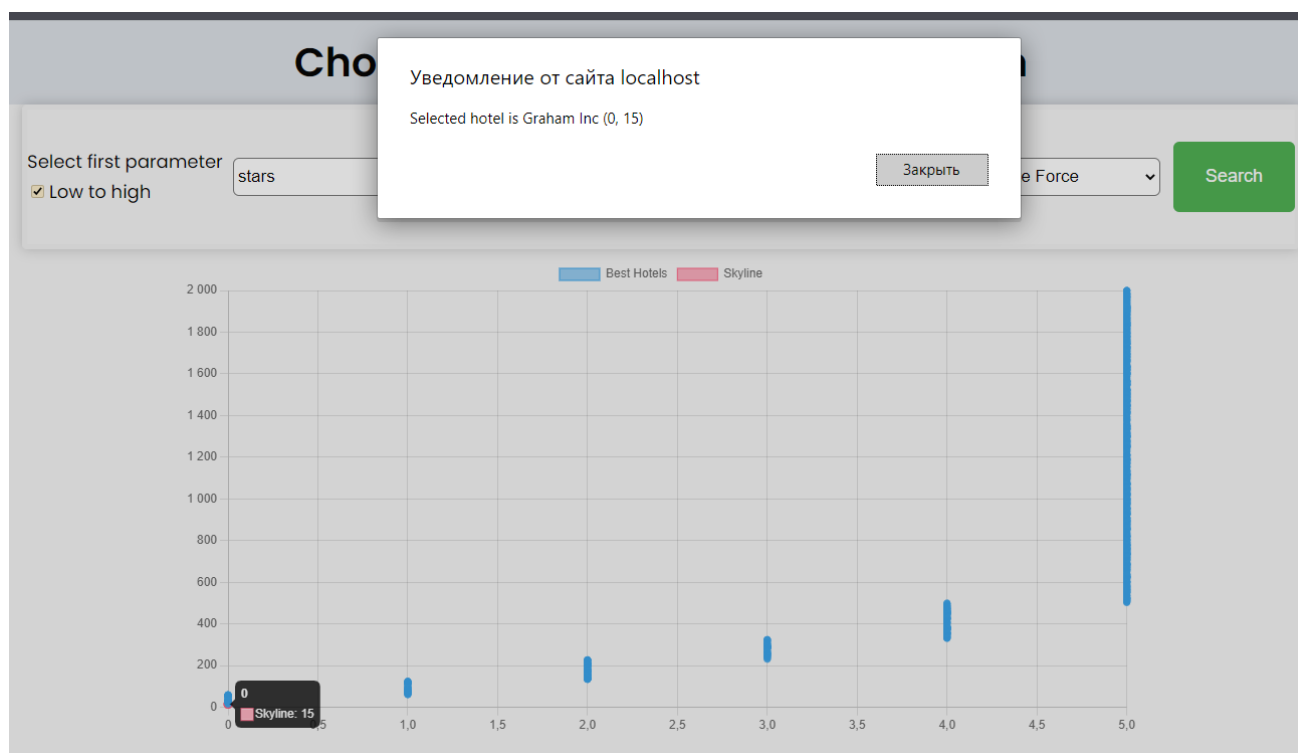
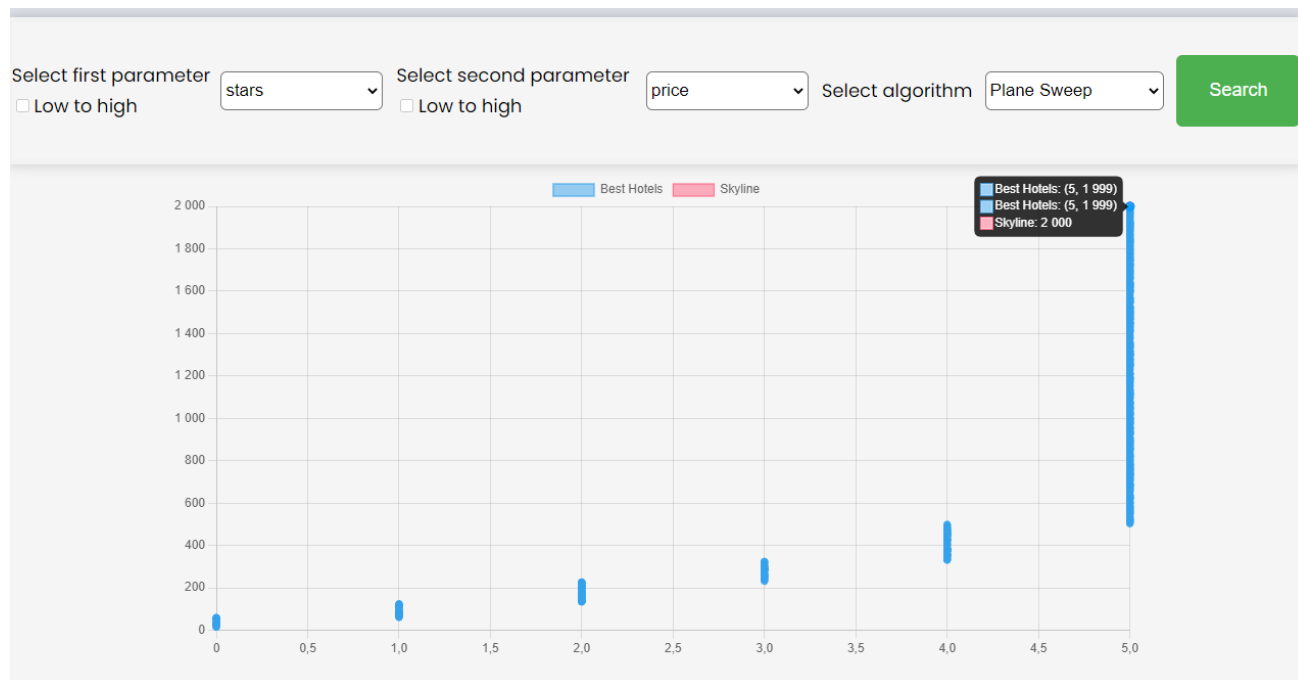
Test all

Search



Algoritmy plane sweep a brute force si s těmito daty vedly nejhůře, zatímco algoritmy BNL a divide and conquer si vedly výrazně lépe.

Takto vypadá skyline pro silně korelovaná data, téměř vždy se jedná o jediný bod.



Пříklad výběru hotelu z grafu

## 6. Diskuze:

Problémy a zlepšení, které lze dále realizovat:

- Implementace rozšířených algoritmů pro vyhledávání hotelů na základě složitějších kritérií.
- Optimalizace algoritmů pro rychlejší výpočet výsledků při velkém množství hotelů.
- Zlepšení uživatelského rozhraní aplikace, např. implementace možnosti filtru hotelů na základě různých kritérií.
- Rozšíření vstupních možností aplikace, např. integrace s API pro získávání aktuálních dat hotelů.

## 7. Závěr:

Na základě provedených experimentů a implementace projektu lze konstatovat, že aplikace splňuje požadavky na vyhledávání hotelů na základě definovaných parametrů. Algoritmy implementované v aplikaci dosahují dobrého výkonu pro různé velikosti vstupních dat a kritérií vyhledávání. Další možné vylepšení by mohlo být provedeno v budoucnu pro zvýšení výkonu a uživatelské přívětivosti.