Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни
«Проектування алгоритмів»

«Неінформативний, інформативний та локальний пошук»

Виконав(ла) ІТ-02 Макаров І.С. _____
(шифр, прізвище, ім'я, по батькові)

Перевірив Головченко М.М. _____
(прізвище, ім'я, по батькові)

Київ 2021

ЗМІСТ

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Мета роботи – розглянути та дослідити алгоритми неінформативного, інформативного та локального пошуку. Провести порівняльний аналіз ефективності використання алгоритмів.

2 ЗАВДАННЯ

Записати алгоритм розв’язання задачі у вигляді псевдокоду, відповідно до варіанту (таблиця 2.1).

Реалізувати програму, яка розв’язує поставлену задачу згідно варіанту (таблиця 2.1) за допомогою алгоритму неінформативного пошуку **АНП**, алгоритму інформативного пошуку **АІП,** що використовує задану евристичну функцію Func, або алгоритму локального пошуку **АЛП та бектрекінгу,** що використовує задану евристичну функцію Func.

Програму реалізувати на довільній мові програмування.

Увага! Алгоритм неінформативного пошуку **АНП,** реалізовується за принципом «AS IS», тобто так, як є, без додаткових модифікацій (таких як перевірка циклів, наприклад).

Провести серію експериментів для вивчення ефективності роботи алгоритмів. Кожний експеримент повинен відрізнятись початковим станом. Серія повинна містити не менше 20 експериментів для кожного алгоритму. За проведеними серіями необхідно визначити:

- середню кількість етапів (кроків), які знадобилось для досягнення розв’язку (ітерації);
- середню кількість випадків, коли алгоритм потрапляв в глухий кут (не міг знайти оптимальний розв’язок) – якщо таке можливе;
- середню кількість згенерованих станів під час пошуку;
- середню кількість станів, що зберігаються в пам’яті під час роботи програми.

Передбачити можливість обмеження виконання програми за часом (30 хвилин) та використання пам’яті (512 Мб)

Використані позначення:

- **8-ферзів** – Задача про вісім ферзів полягає в такому розміщенні восьми ферзів на шахівниці, що жодна з них не ставить під удар один одного. Тобто, вони не повинні стояти в одній вертикалі, горизонталі чи діагоналі.

- **8-puzzle** – гра, що складається з 8 однакових квадратних пластинок з нанесеними числами від 1 до 8. Пластинки поміщаються в квадратну коробку, довжина сторони якої в три рази більша довжини сторони пластинок, відповідно в коробці залишається незаповненим одне квадратне поле. Мета гри – переміщаючи пластинки по коробці досягти впорядковування їх по номерах, бажано зробивши якомога менше переміщень.
- **LDFS** – Пошук вглиб з обмеженням глибини.
- **BFS** – Пошук вшир.
- **IDS** – Пошук вглиб з ітеративним заглибленням.
- **A*** – Пошук A*.
- **RBFS** – Рекурсивний пошук за першим найкращим співпадінням.
- **F1** – кількість пар ферзів, які б’ють один одного з урахуванням видимості (ферзь A може стояти на одній лінії з ферзем B, проте між ними стоїть ферзь C; тому A не б’є B).
- **F2** – кількість пар ферзів, які б’ють один одного без урахування видимості.
- **H1** – кількість фішок, які не стоять на своїх місцях.
- **H2** – Манхетенська відстань.
- **COLOR** – Задача розфарбування карти самостійно обраної країни, не менше 20 регіонів (областей). Необхідно розфарбувати карту не більше ніж у 4 різні кольори. Мається на увазі приписування кожному регіону власного кольору так, щоб кольори сусідніх регіонів відрізнялись. Використовувати евристичну функцію, яка повертає кількість пар суміжних вузлів, що мають однаковий колір (тобто кількість конфліктів). Реалізувати алгоритм пошуку із поверненнями (backtracking) для розв’язання поставленої задачі. Для підвищення швидкодії роботи алгоритму використати евристичну функцію, а початковим станом вважати випадкову вершину.

- **HILL –** Пошук зі сходженням на вершину з використанням із використанням руху вбік (на 100 кроків) та випадковим перезапуском (кількість необхідних разів запуску визначити самостійно).
- **ANNEAL** – Локальний пошук із симуляцією відпалу. Робоча характеристика – залежність температури T від часу роботи алгоритму t. Можна розглядати лінійну залежність: $T = 1000 - k \cdot t$, де k – змінний коефіцієнт.
- **BEAM** – Локальний променевий пошук. Робоча характеристика – кількість променів k. Експерименти проводи із кількістю променів від 2 до 21.
- **MRV** – евристика мінімальної кількості значень;
- **DGR** – ступенева евристика.

ВИКОНАННЯ

2.1 Псевдокод алгоритмів

Код IDS

```
ids (root, goal){
depth = 0
repeat {
      result = DLS(root, goal, depth)
      if result is solution
            return result

      depth += 1
}
}
dls (node, goal, depth) {
if depth == 0 and node == goal
      return node
else if depth > 0
      for each child in expand(node)
            dls(child, goal, depth — 1)
else
      return no-solution
}
```

Код RBFS

```
function Recursive-Best-First-Search(problem) {
   return RBFS(problem, InitialState(problem) , ∞)
}

function RBFS(problem, node, f_limit) {
   if node is solution
```

```
 return node
   successors ← Expand(node, problem)
   if not successors
    then return failure, ∞

   for each s in successors do
     f[s] ← max(g(s)+h(s) , f[node] )

repeat
    best ← вузол з найменшим f-значенням в множині successors
    if f[best] > f_limit then return failure, f[best]
    alternative ← друге після найменшого f-значення в множині successors
    result, f[best] ← RBFS(problem, best, min{f_limit, alternative))

    if result ≠ failure
 return result
}
```

2.2 Програмна реалізація

2.2.1 Вихідний код

https://github.com/kinfi4/AlgorithmsDataStructures/tree/main/Algorithms/8-queen%20problem

2.2.2 Приклади роботи

На рисунках 3.1 і 3.2 показані приклади роботи програми для різних алгоритмів пошуку.

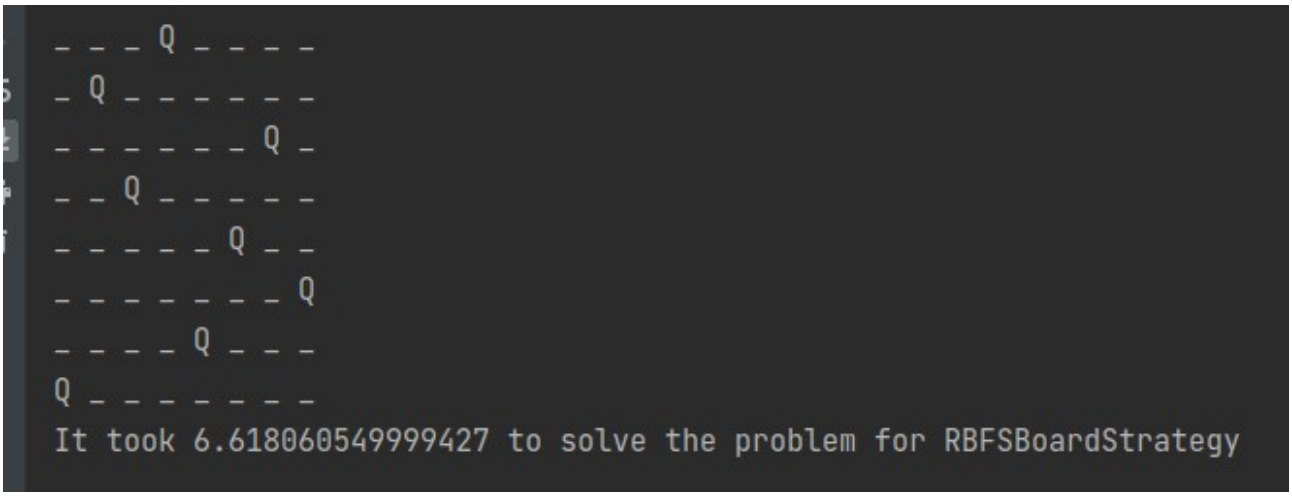

Рисунок 3.1 – Алгоритм IDS

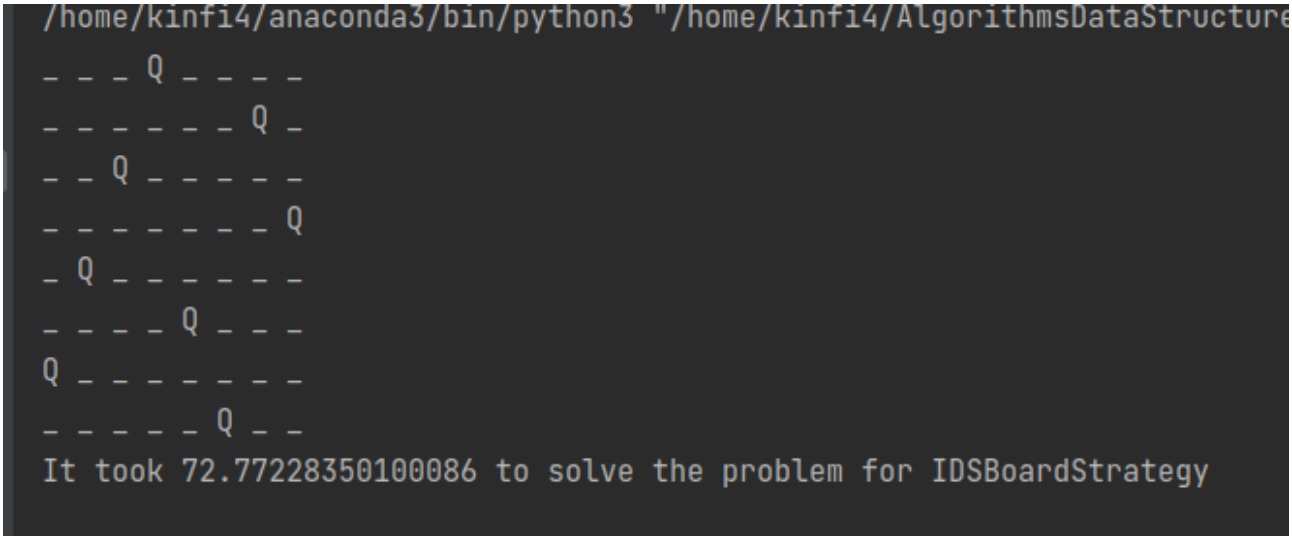

Рисунок 3.2 – Алгоритм RBFS

2.3 Дослідження алгоритмів

В таблиці 3.1 наведені характеристики оцінювання алгоритму RBFS ,задачі 8 queen для 20 початкових станів.

Таблиця 3.1 – Характеристики оцінювання алгоритму RBFS

Початкові стани	Ітерації	Всього станів	Час виконання
Стан 1	555	13277	1.3
Стан 2	4	404	0.03
Стан 3	5851	60257	5.7
Стан 4	5	532	0.04
Стан 5	4	405	0.03
Стан 6	15414	174443	14.8
Стан 7	145147	1351423	120
Стан 8	16941	172232	16
Стан 9	15474	170868	15
Стан 10	3163	34001	2
Стан 11	84551	841054	79
Стан 12	299508	2835878	228
Стан 13	86848	493789	43
Стан 14	3108131	12352343	1288
Стан 15	737	21559	1.7
Стан 16	2762295	14151241	1615
Стан 17	2297133	13205781	956
Стан 18	55118	584213	46
Стан 19	4573	48759	3
Стан 20	5	533	0.03

В таблиці 3.2 наведені характеристики оцінювання алгоритму IDS, задачі 8 queen для 20 початкових станів.

Таблиця 3.3 – Характеристики оцінювання IDS

Початкові стани	Ітерації	Всього станів	Час виконання
Стан 1	112	23221	0.02
Стан 2	301	45341	0.04
Стан 3	42343	6345568	38
Стан 4	3245	643285	0.5
Стан 5	105	348	0.01
Стан 6	258233	87589365	515
Стан 7	96	437	0.01
Стан 8	38532	5234432	44
Стан 9	239472	82695758	290
Стан 10	394	2345	0.04
Стан 11	74592	2487588	48
Стан 12	98	428	0.01
Стан 13	412	847	0.05
Стан 14	873573	92934751	1251
Стан 15	18243	93763	24
Стан 16	653	2582	0.03
Стан 17	22934	7523925	12
Стан 18	77529	8274523	32
Стан 19	289	3978	0.05
Стан 20	76537	81625	3

ВИСНОВОК

Таааак, ахаха, сподіваюсь тут можна трохи дозволити собі трохи творчості. Я пожалуй напишу на русском, надо будет переписать на украинский, вы скажите. Суть лабораторной работы заключалась в использовании алгоритмов информативного и не информативного поиска для решения задачи 8 ферзей. Не информативный алгоритм у меня был IDS (итеративный поиск в глубину), информативный RBFS (рекурсивный поиск по первому лучшему совпадению). И что я вам могу сказать. Для начала IDS показал себя оч плохо, что так то не удивительно, учитывая что это просто слепой перебор, пускай и не самый плохой, но все же перебор. Ну и для такой NP трудной задачи это сами понимаете не оч. RBFS в целом мне понравился, сам алгоритм не много сложнее IDS, так как надо еще реализовать эвристическую функцию оценки, но зато прирост производительности очевиден, потому что мы уже ходим наугад, а че то там себе видим, че то знаем, как минимум, не рассматриваем откровенно кринжевые ситуации.

КРИТЕРІЇ ОЦІНЮВАННЯ

За умови здачі лабораторної роботи до 25.09.2021 включно максимальний бал дорівнює – 5. Після 25.09.2021 максимальний бал дорівнює – 1.

Критерії оцінювання у відсотках від максимального балу:

- псевдокод алгоритму – 10%;
- програмна реалізація алгоритму – 60%;
- дослідження алгоритмів – 25%;
- висновок – 5%.