**LAPORAN TUGAS KECIL III**

**IF2211 STRATEGI ALGORITMA**

**Penyelesaian Permainan Word Ladder Menggunakan Algoritma UCS, Greedy Best First Search, dan A\***

****

**Disusun oleh**

**Francesco Michael Kusuma**

**13522038**

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika**

**Institut Teknologi Bandung**

**2023/2024**

BAGIAN I

ANALISI DAN IMPLEMENTASI

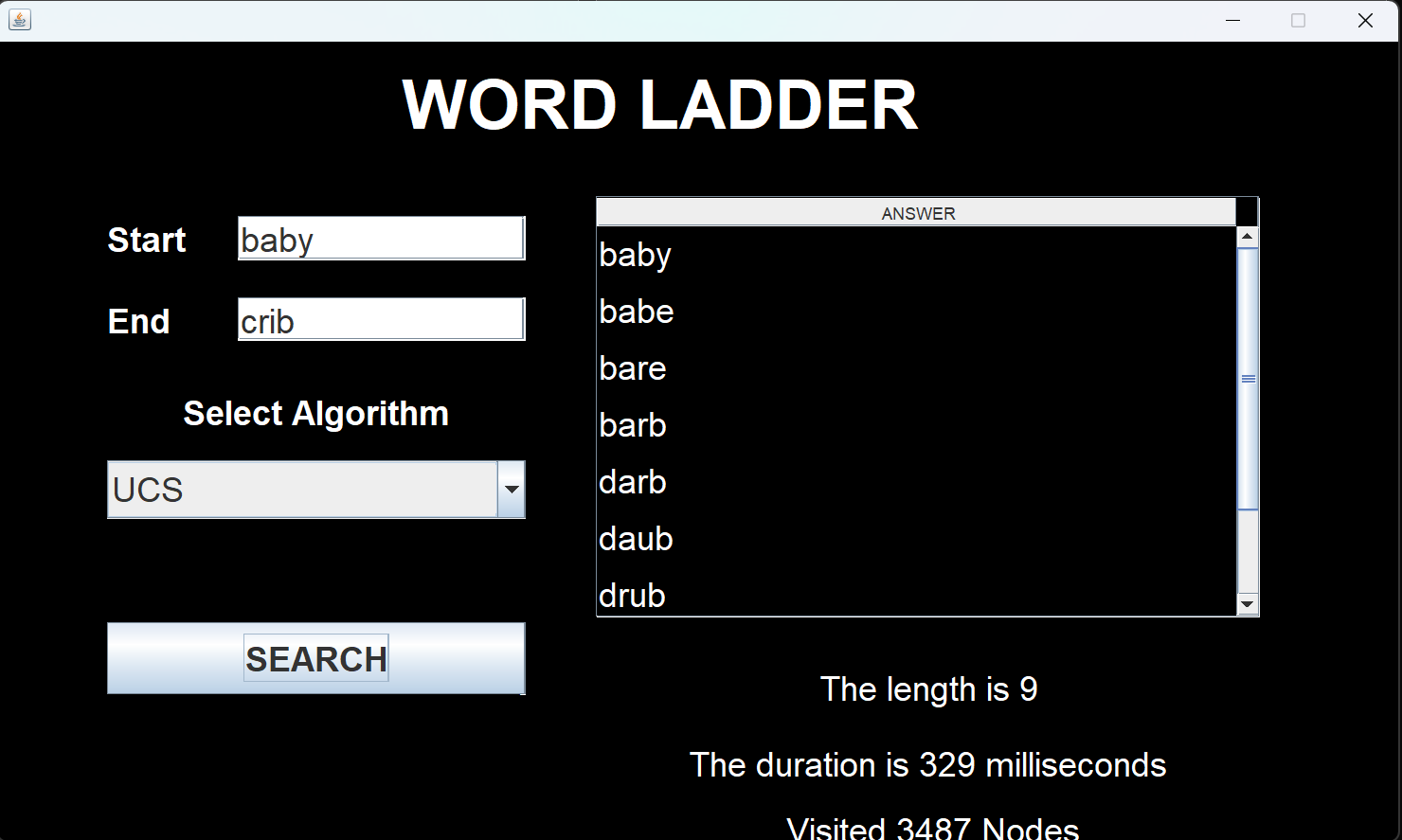
BAGIAN II

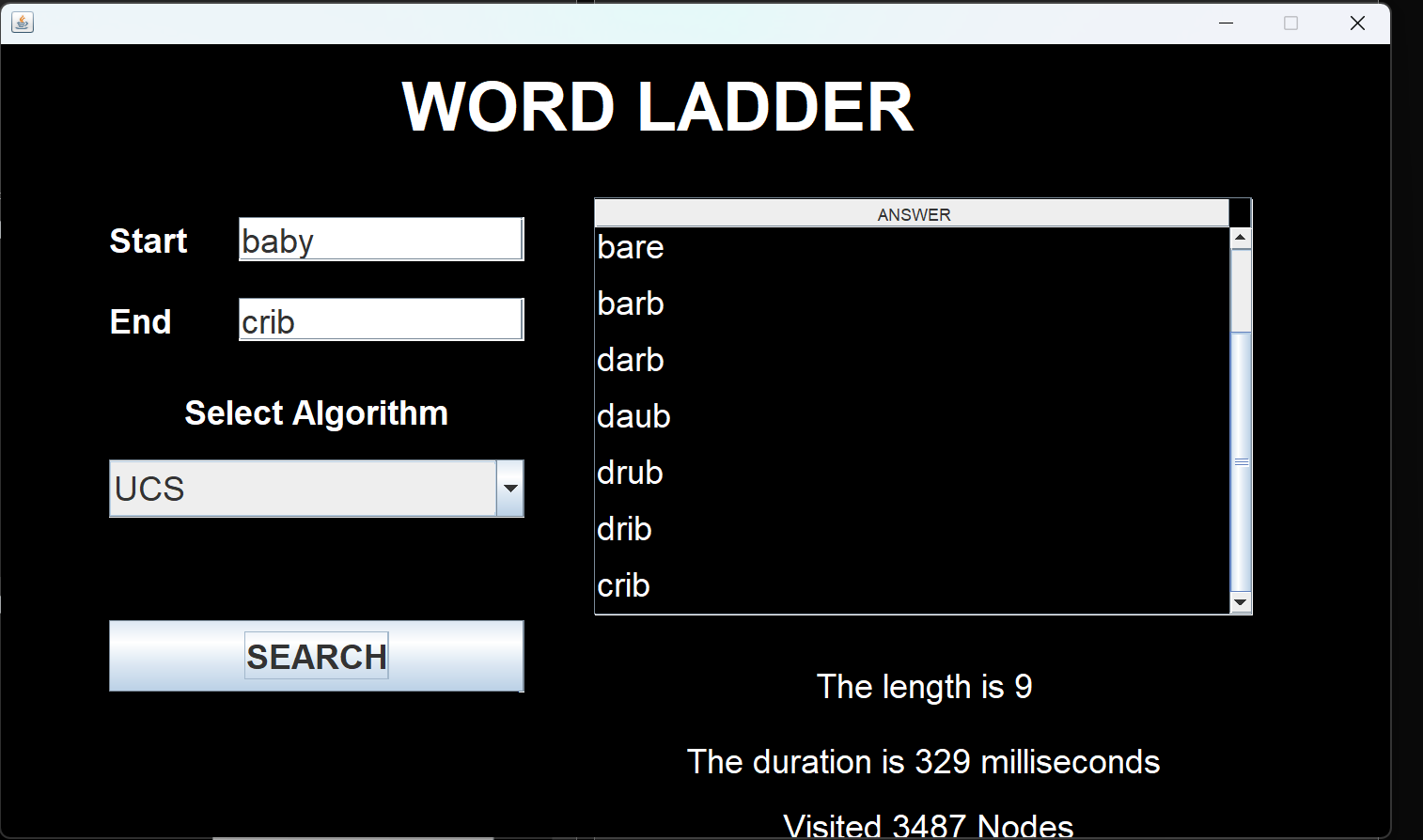
SOURCE CODE

**BAGIAN III**

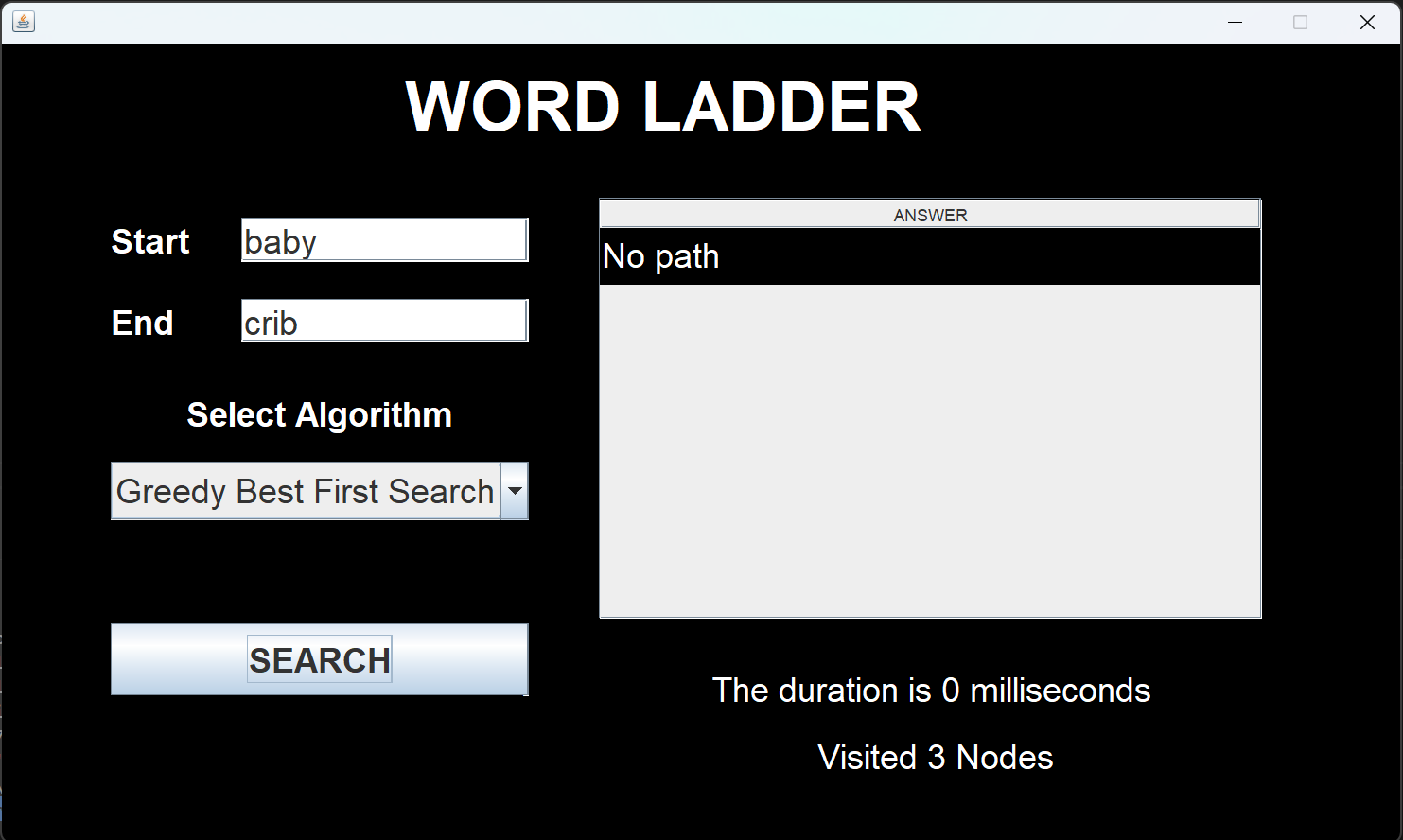
**HASIL UJI COBA**

1. Baby – Crib
2. UCS

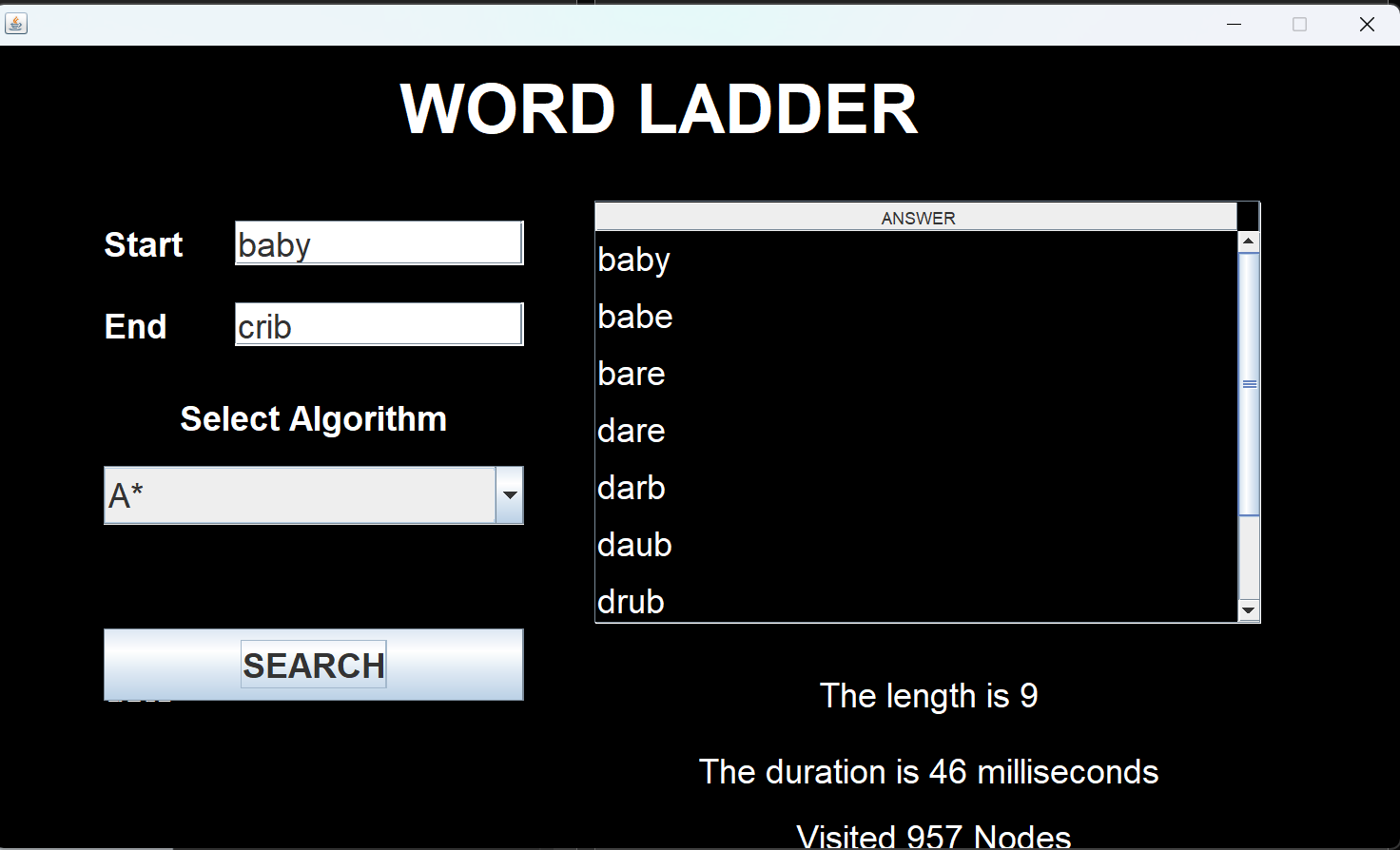


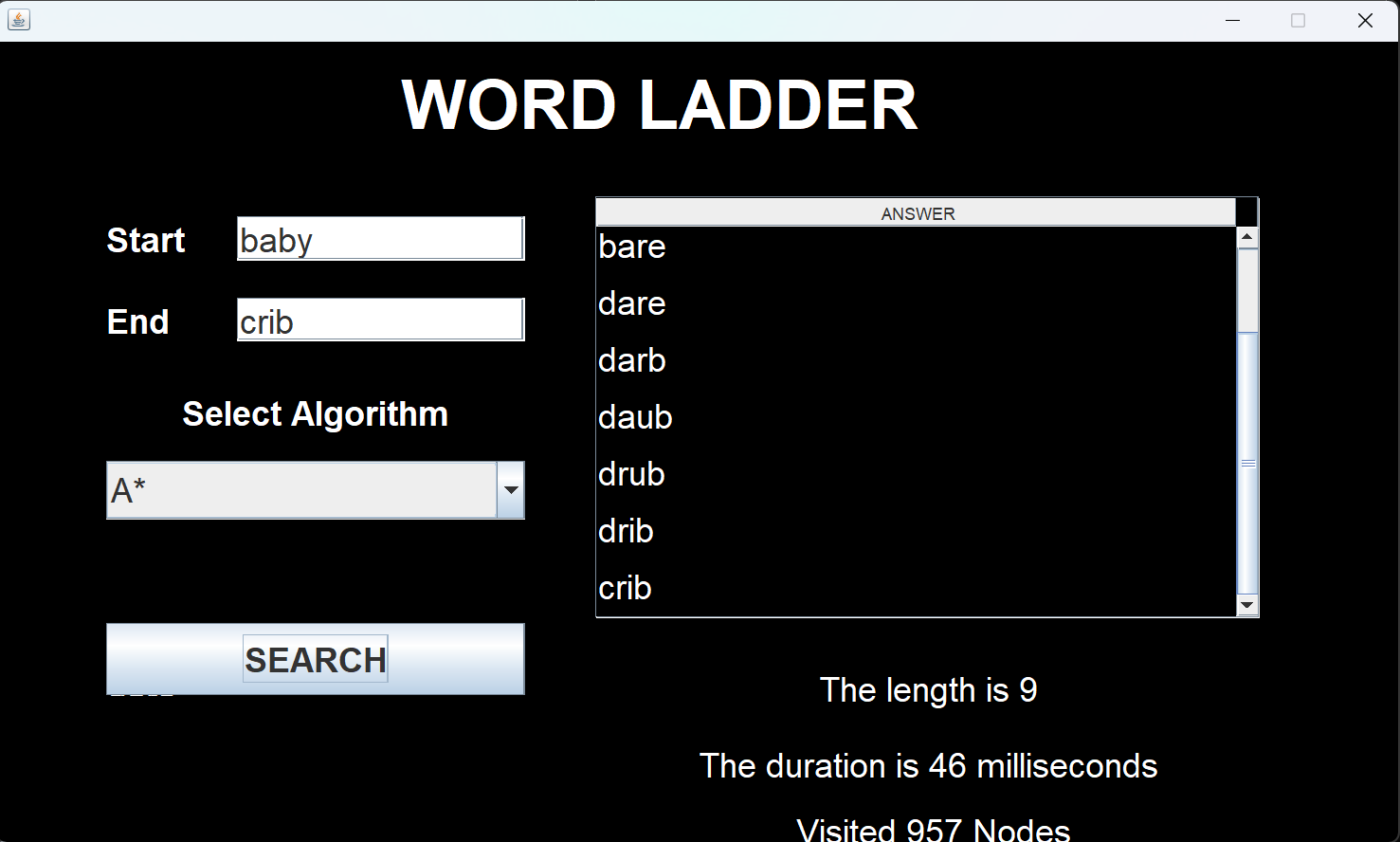


1. GBFS

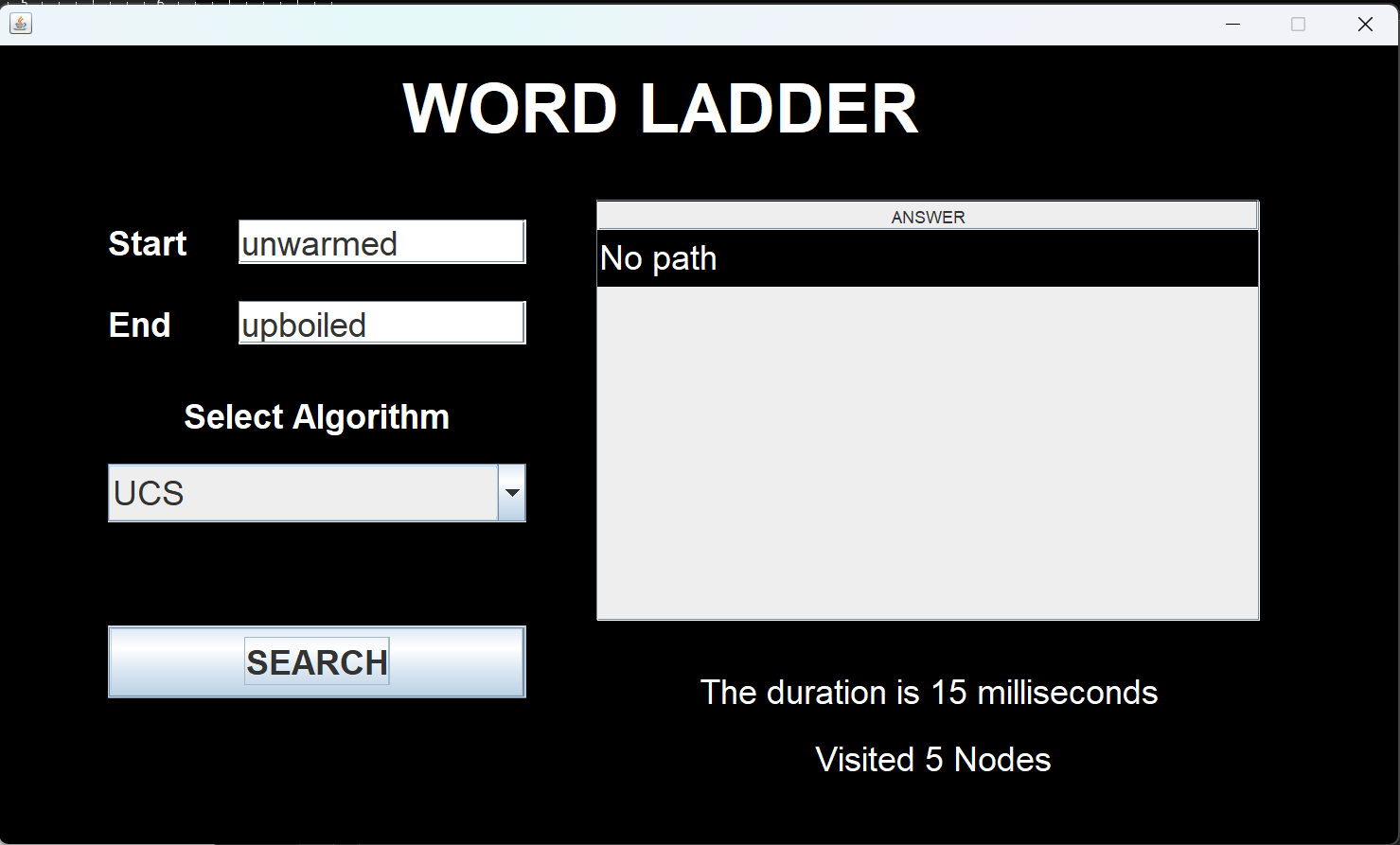


1. A Star

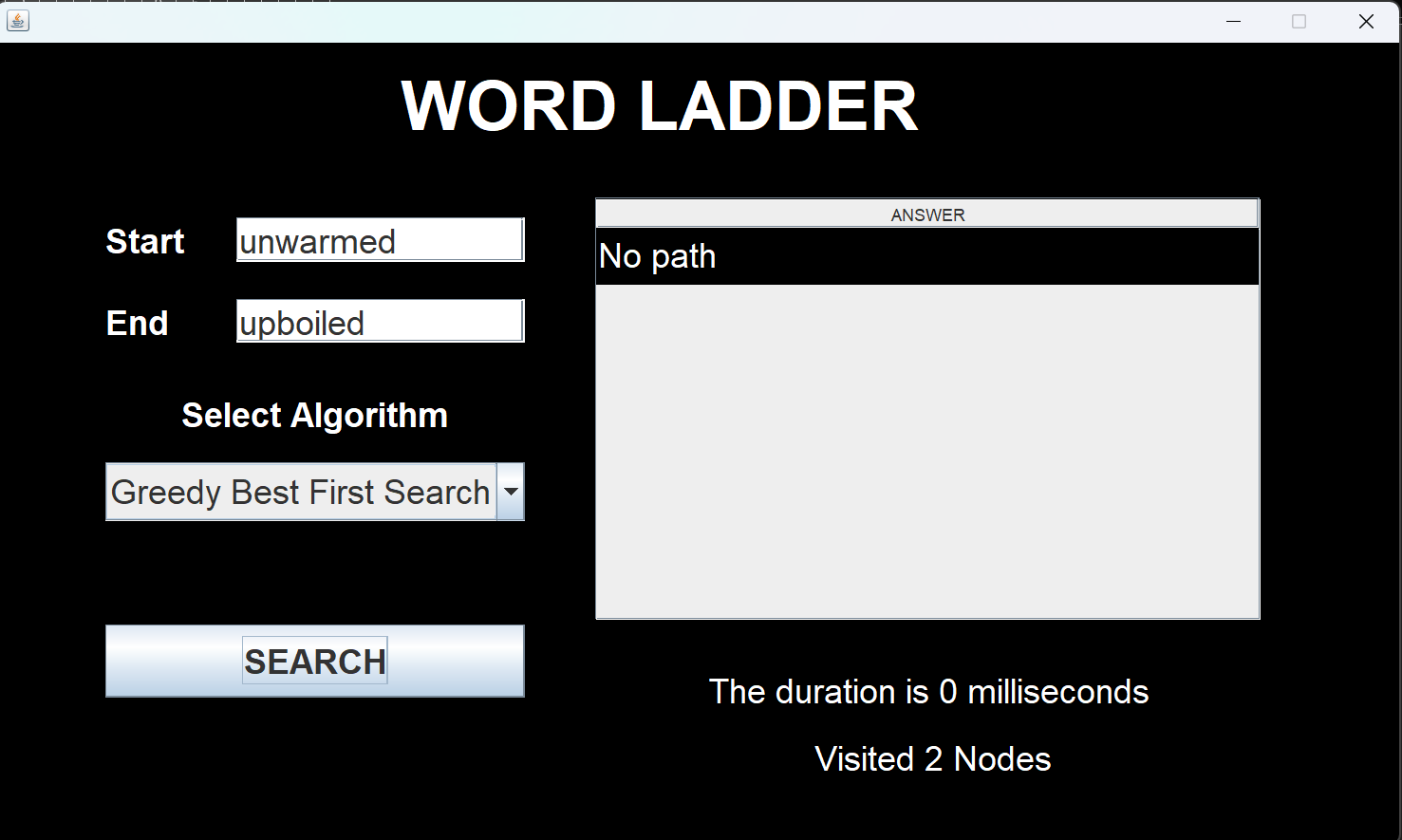




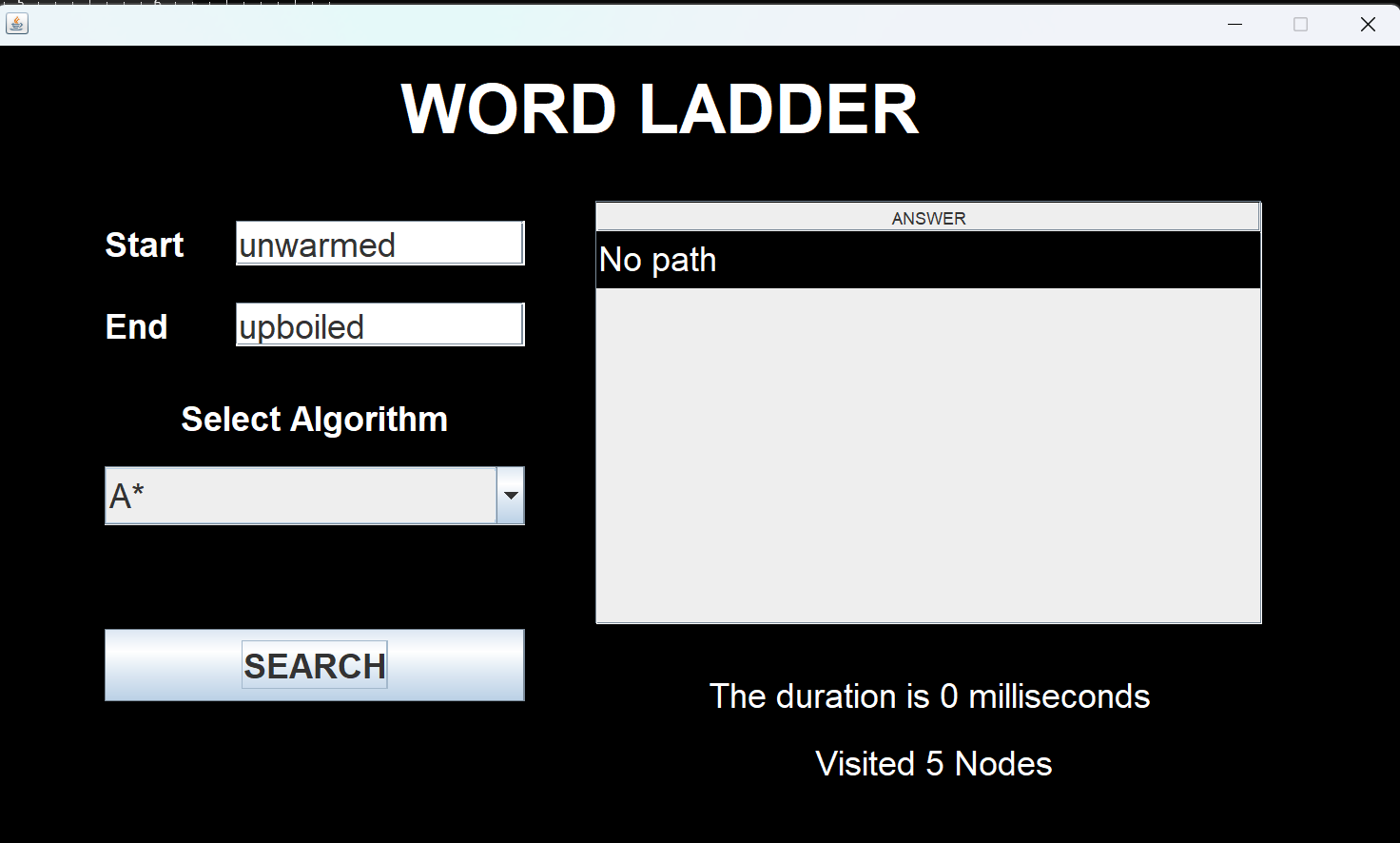
1. Unwarmed – Upboiled
2. UCS



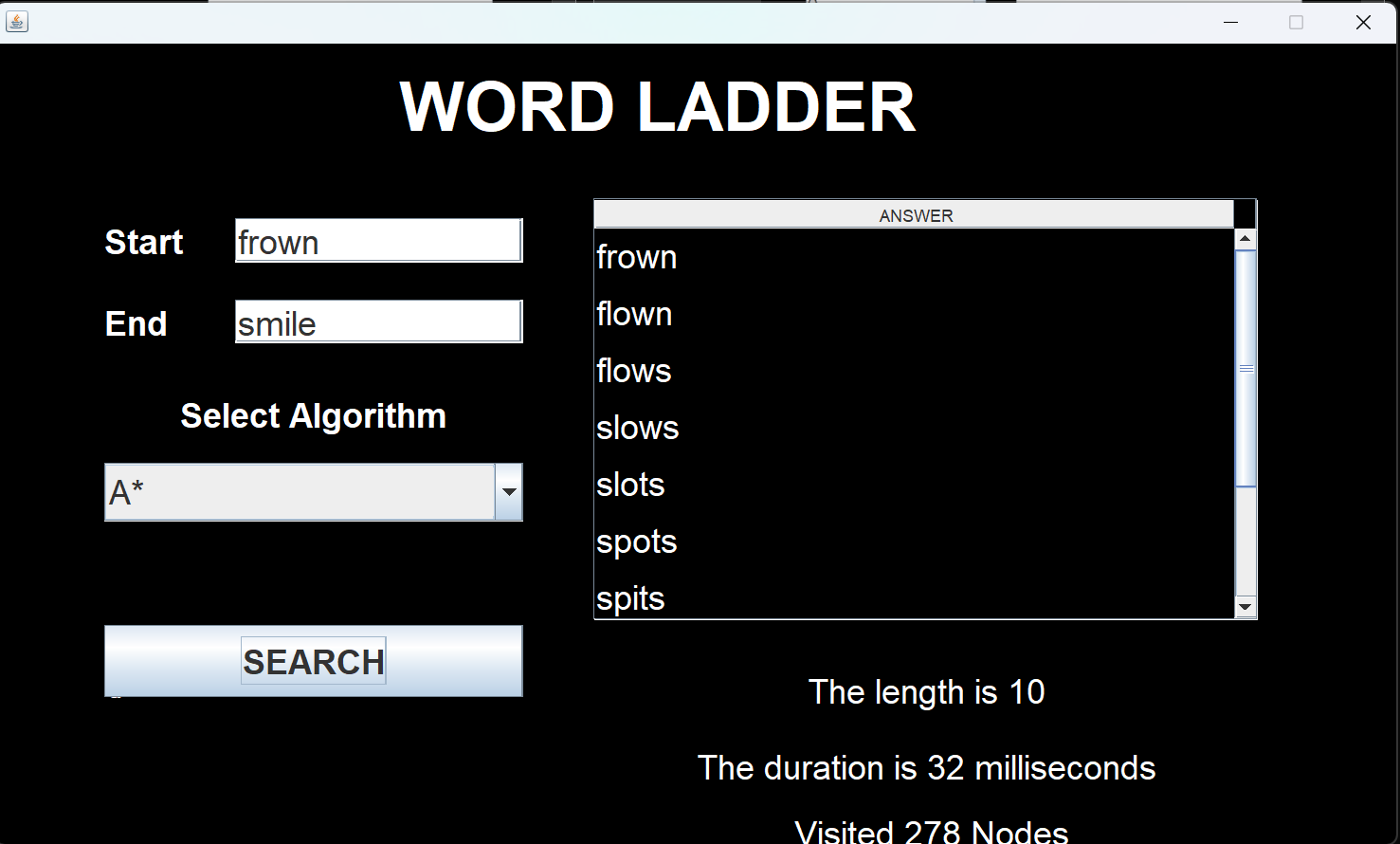
1. GBFS

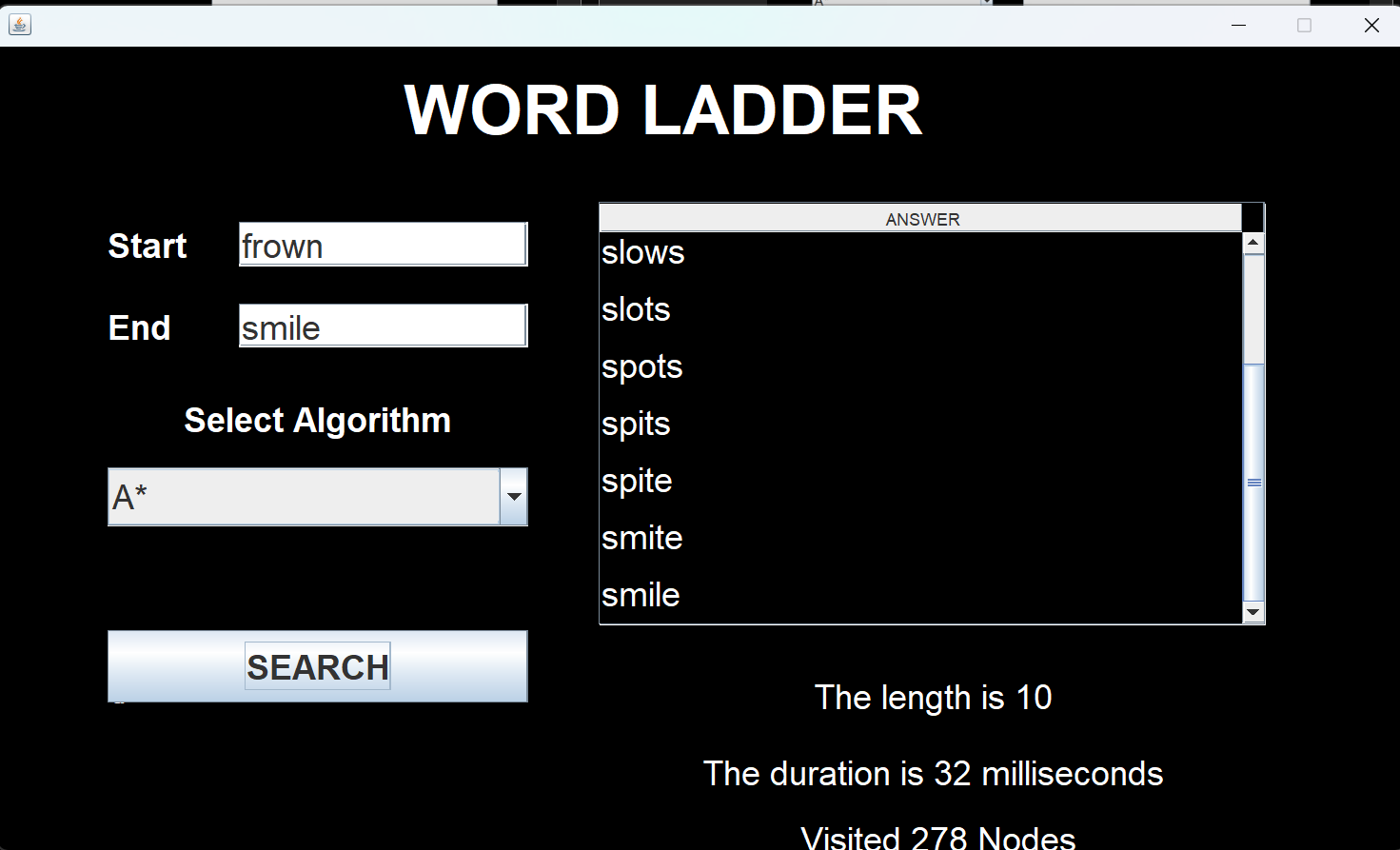


1. A Star

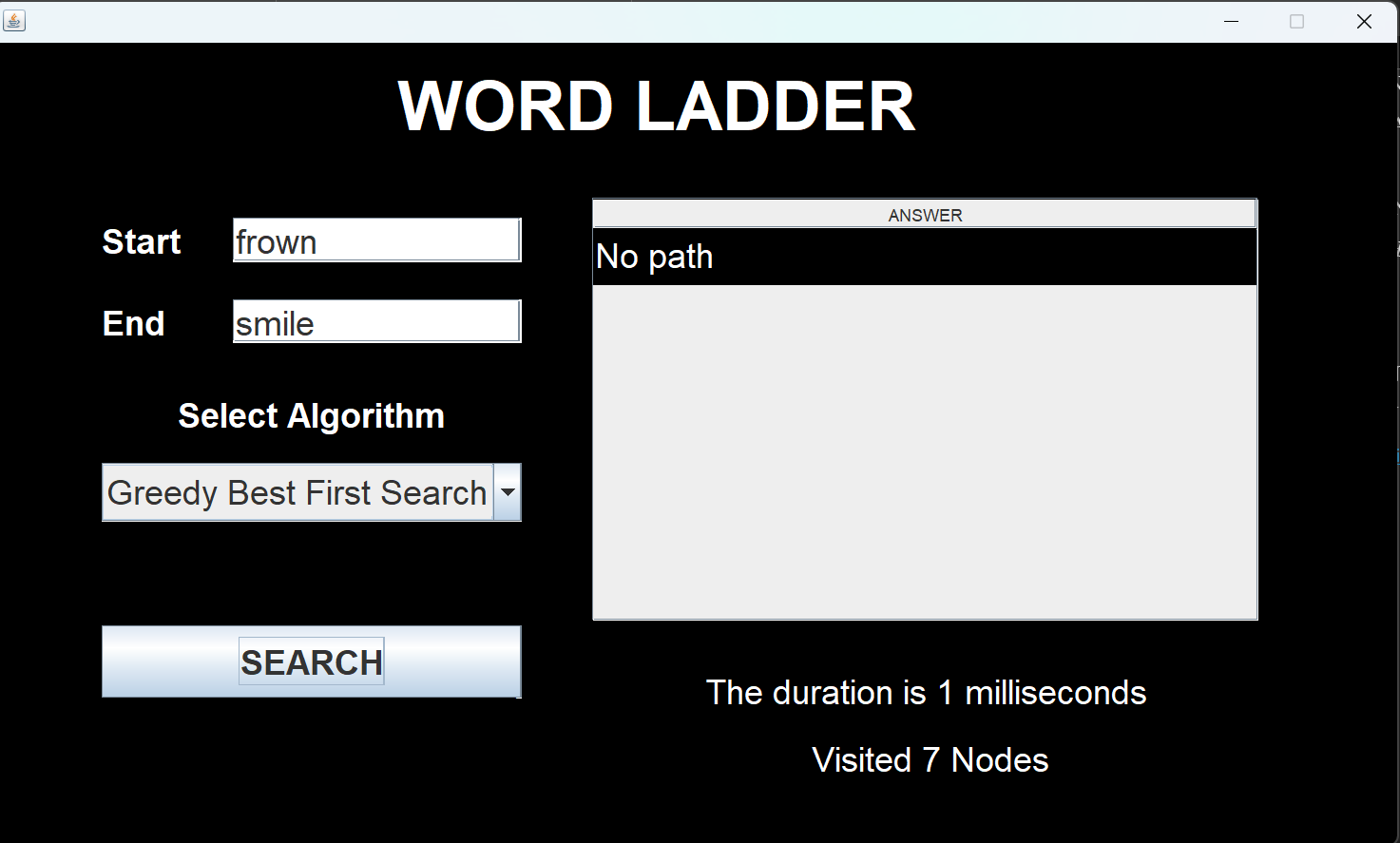


1. Frown – Smile
2. UCS

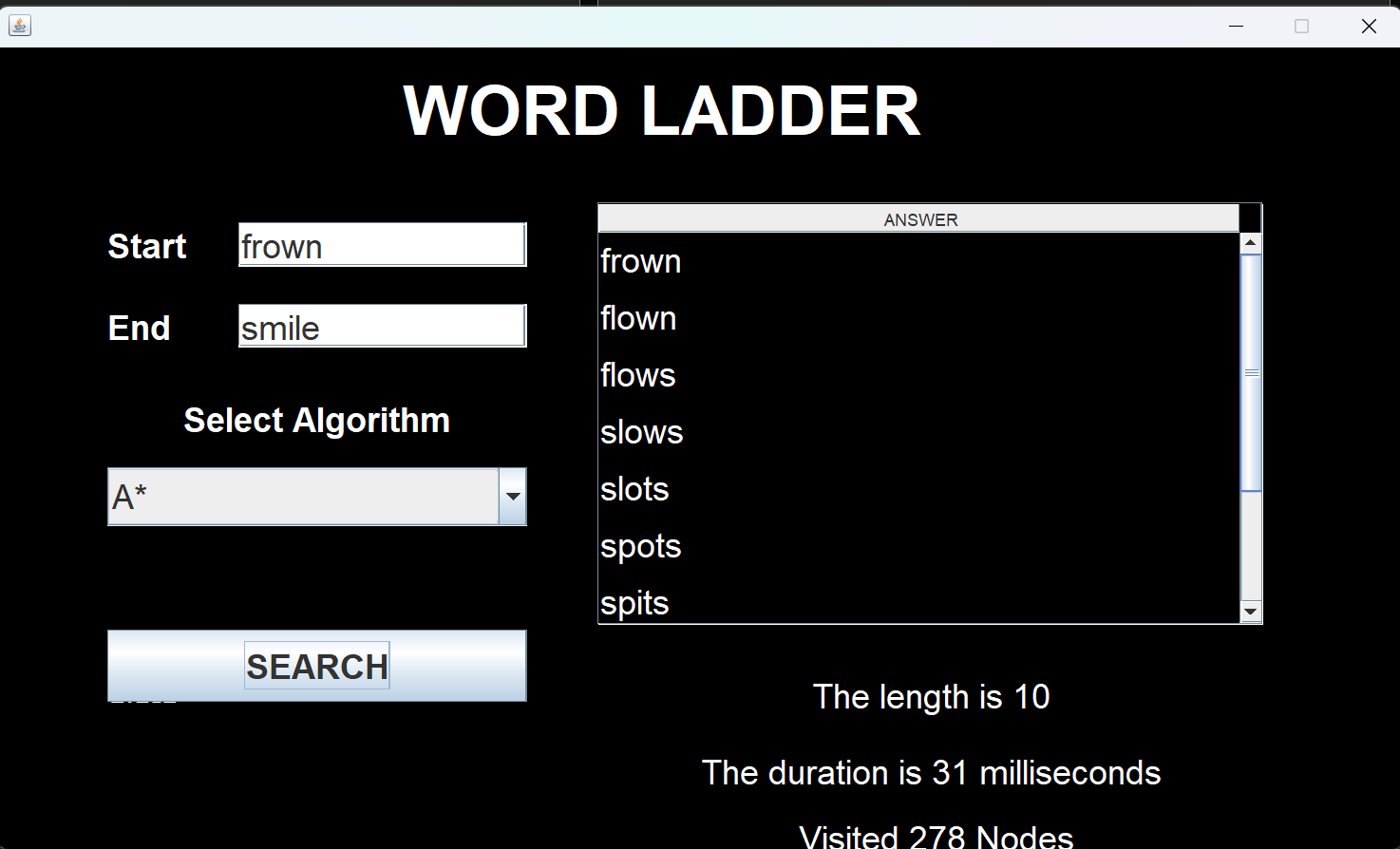


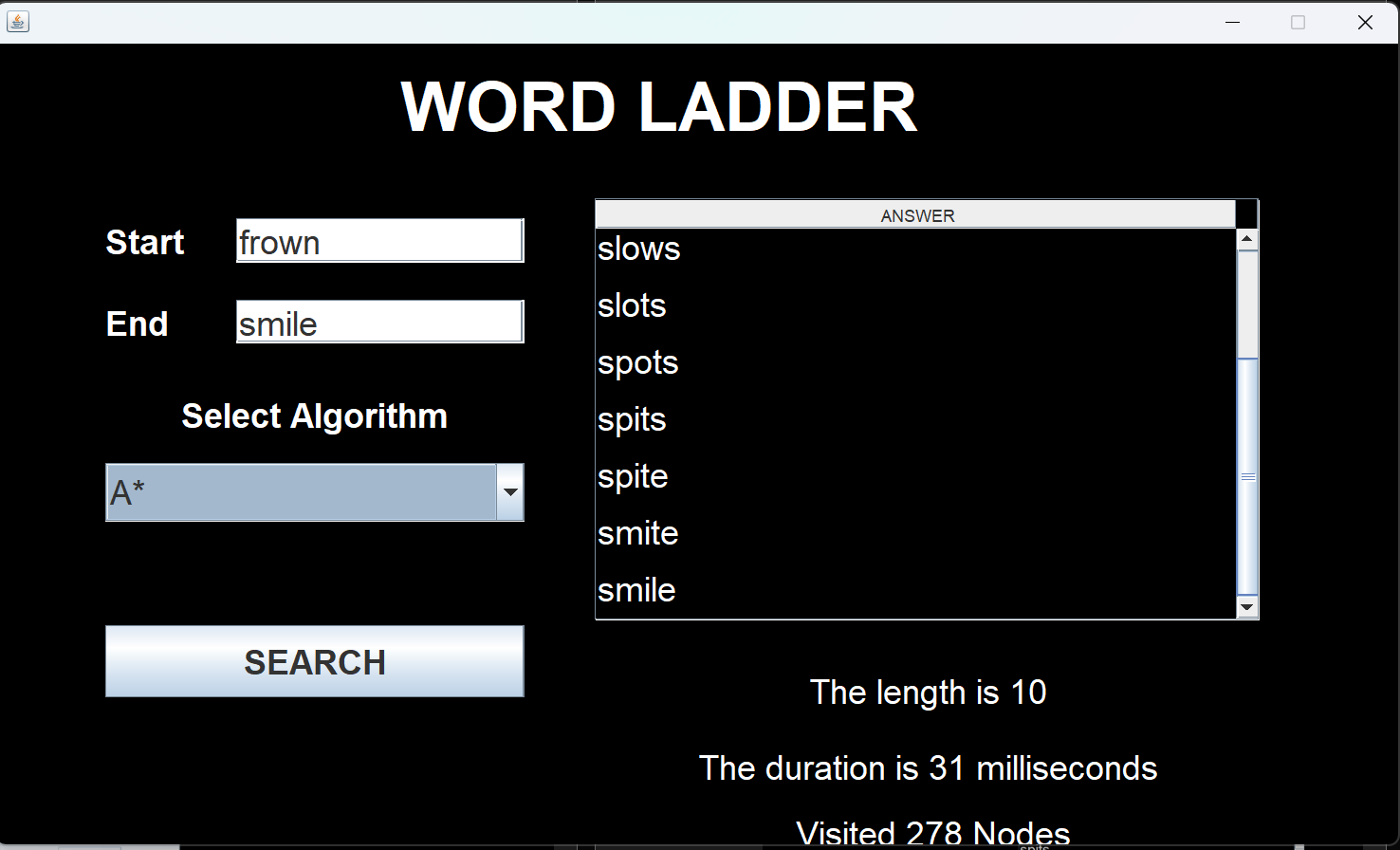


1. GBFS

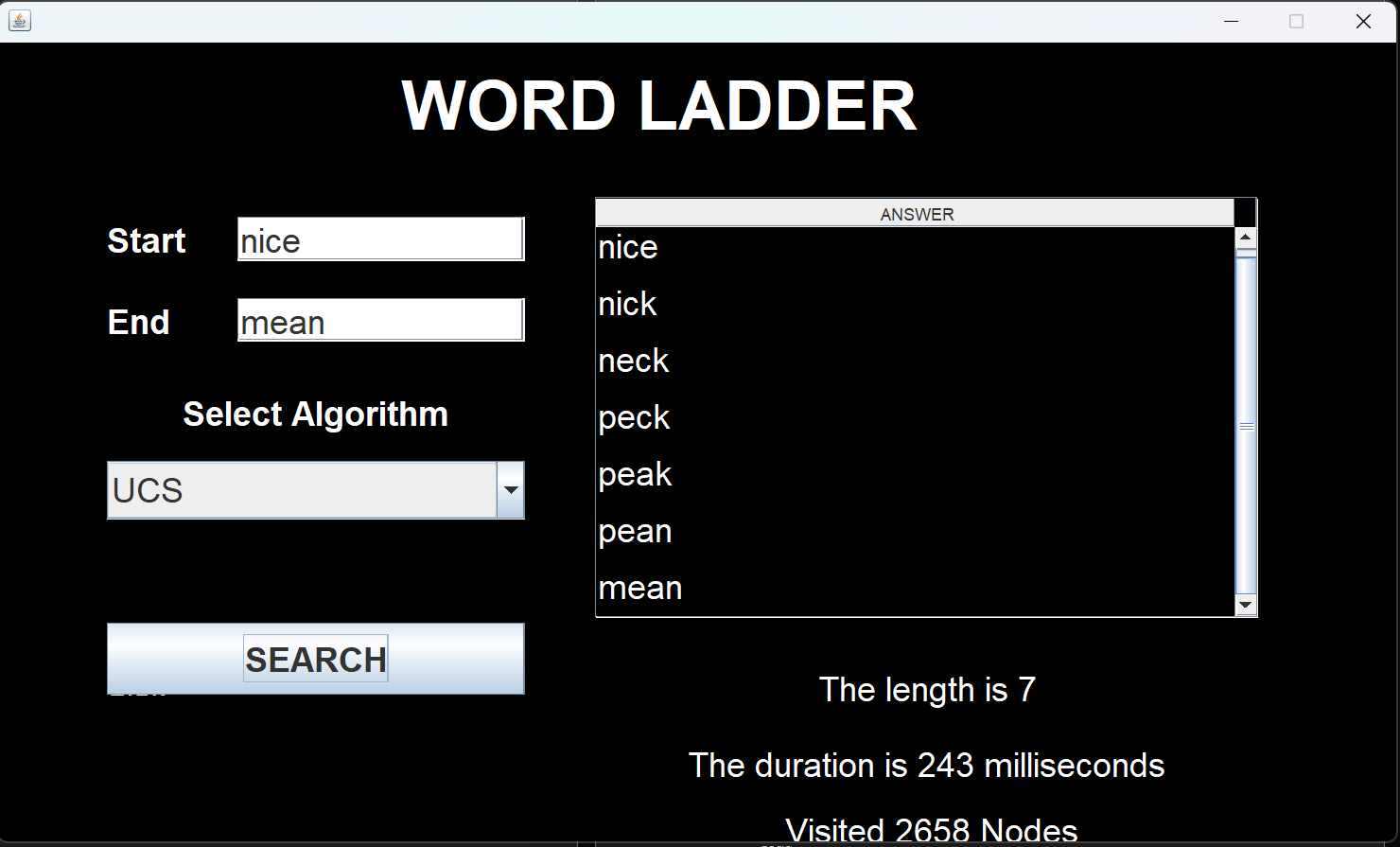


1. A Star

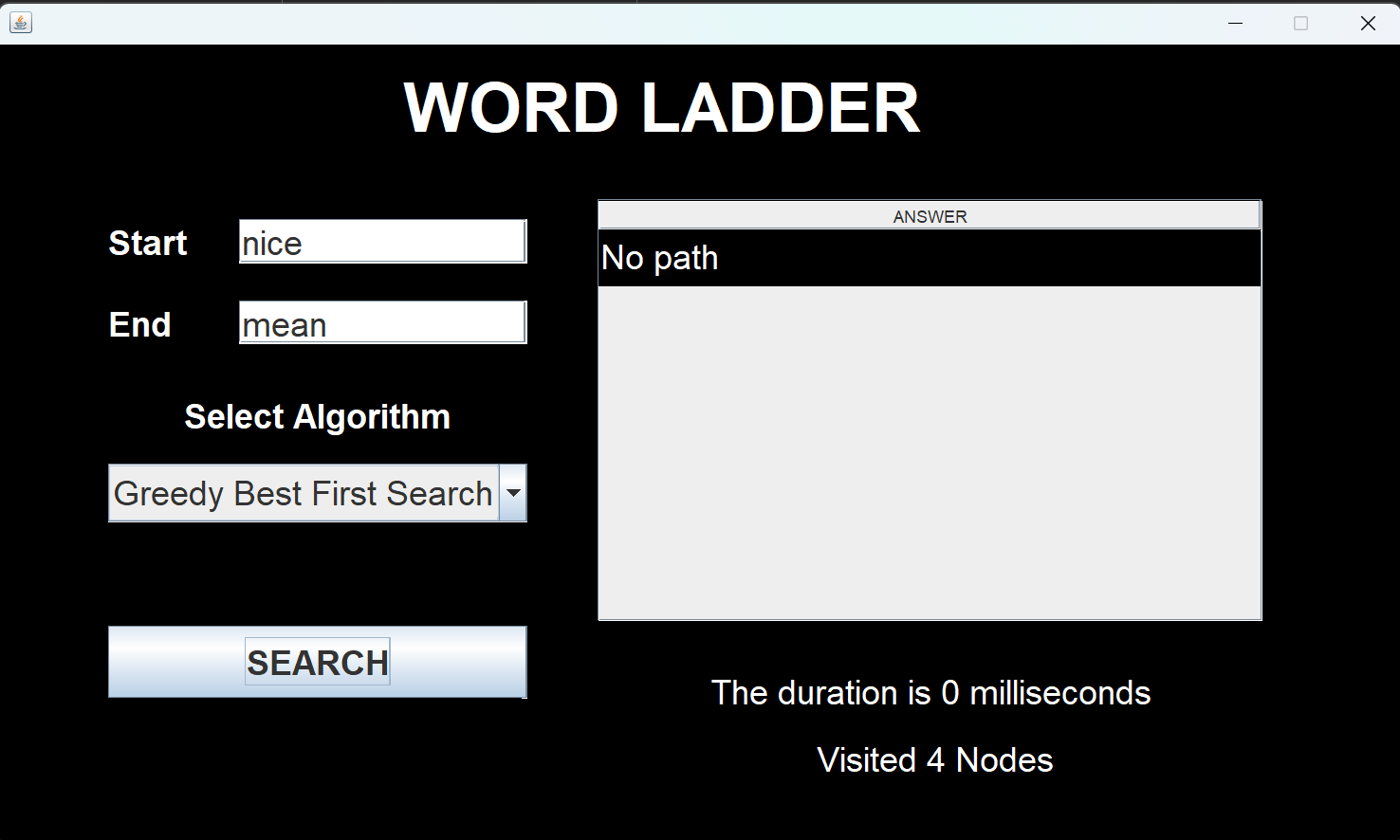




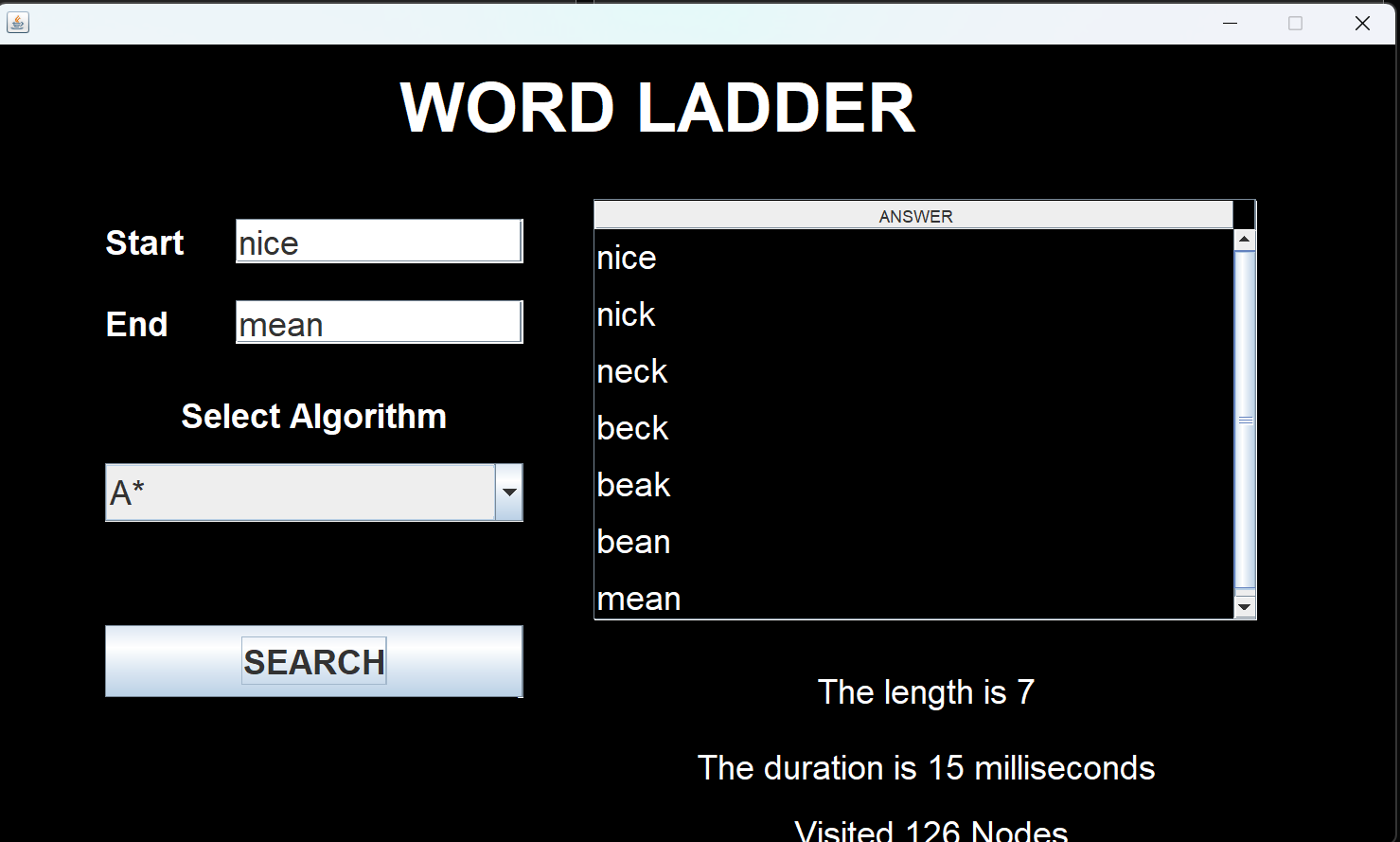
1. Nice – Mean
2. UCS



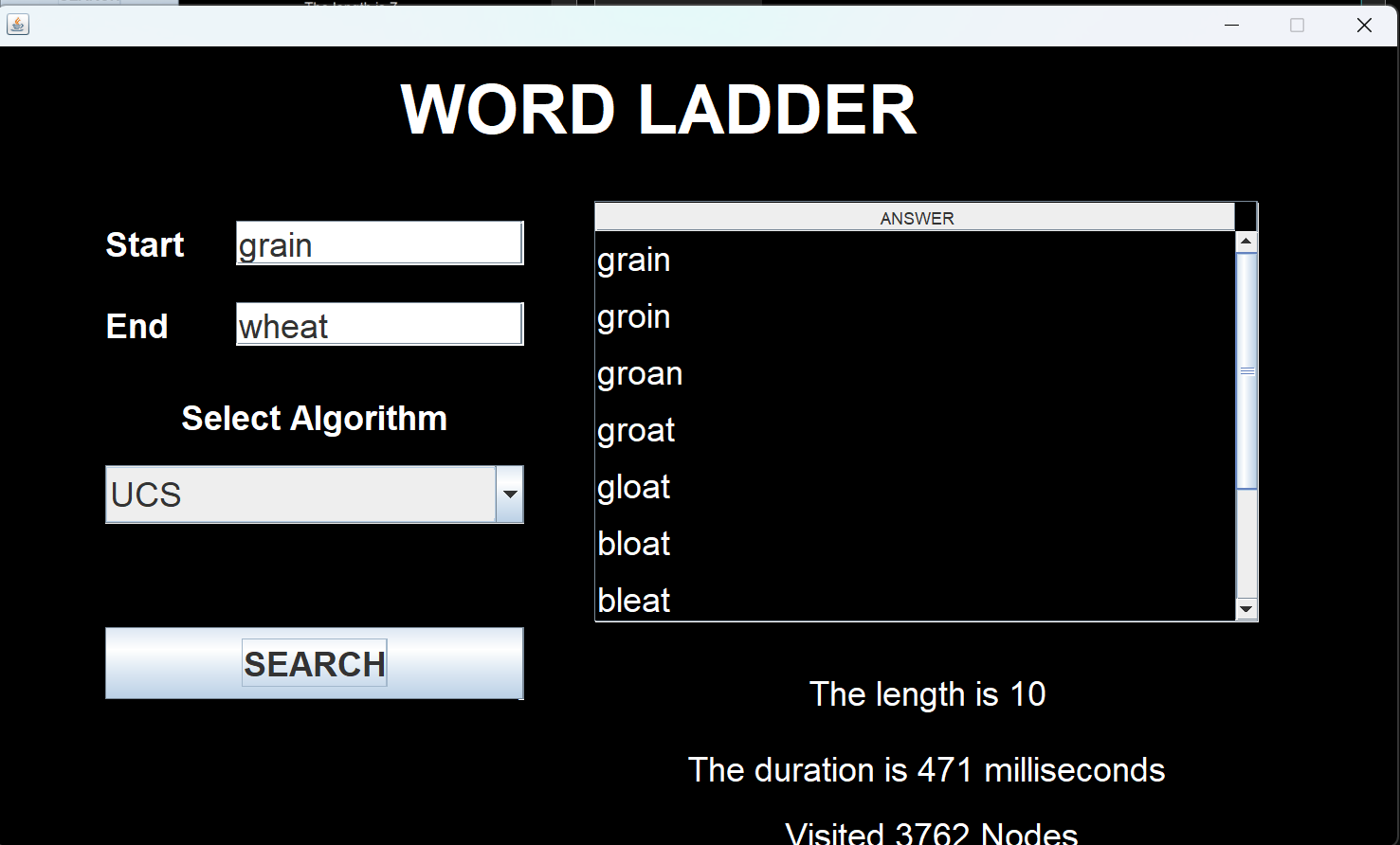
1. GBFS

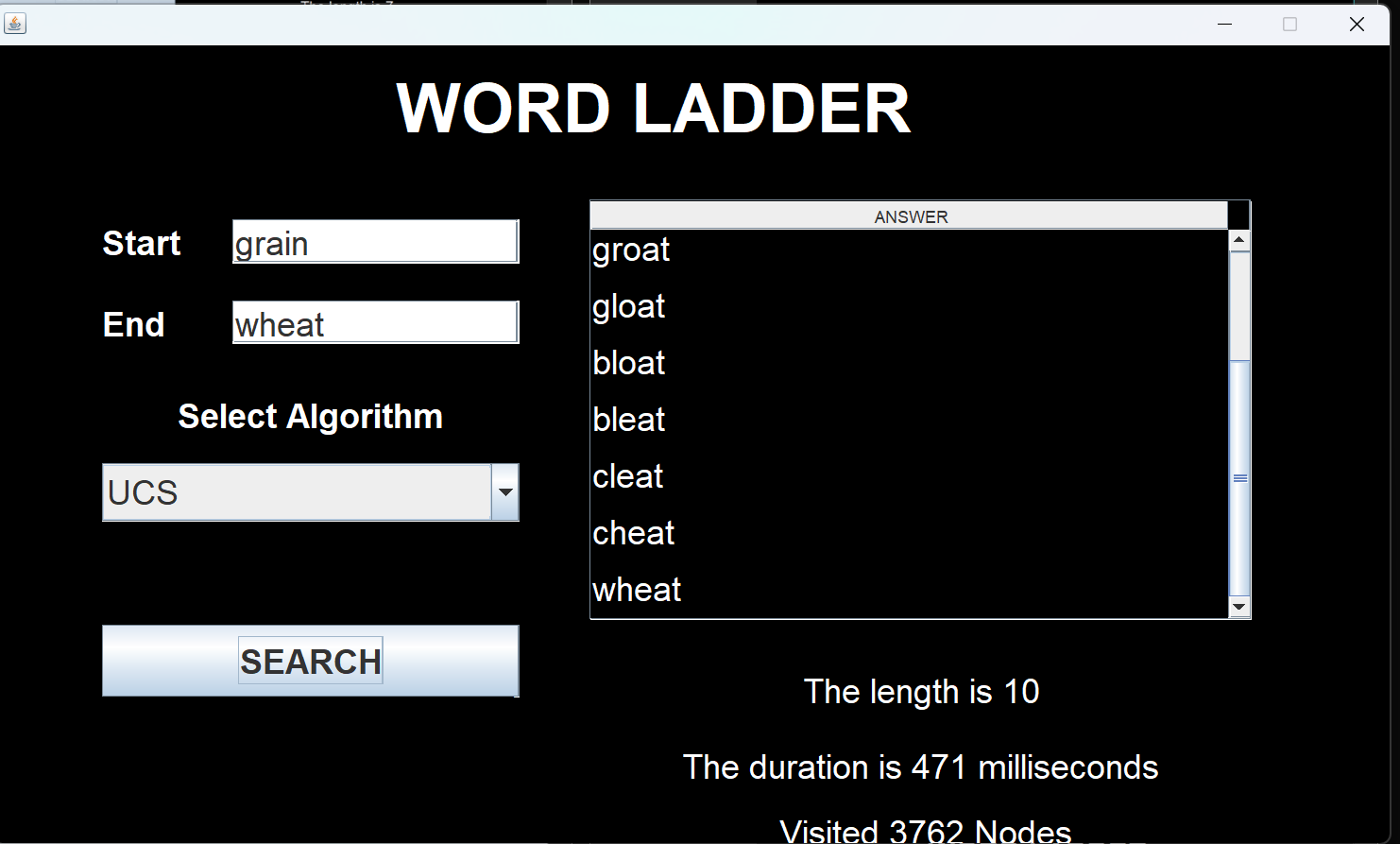


1. A Star

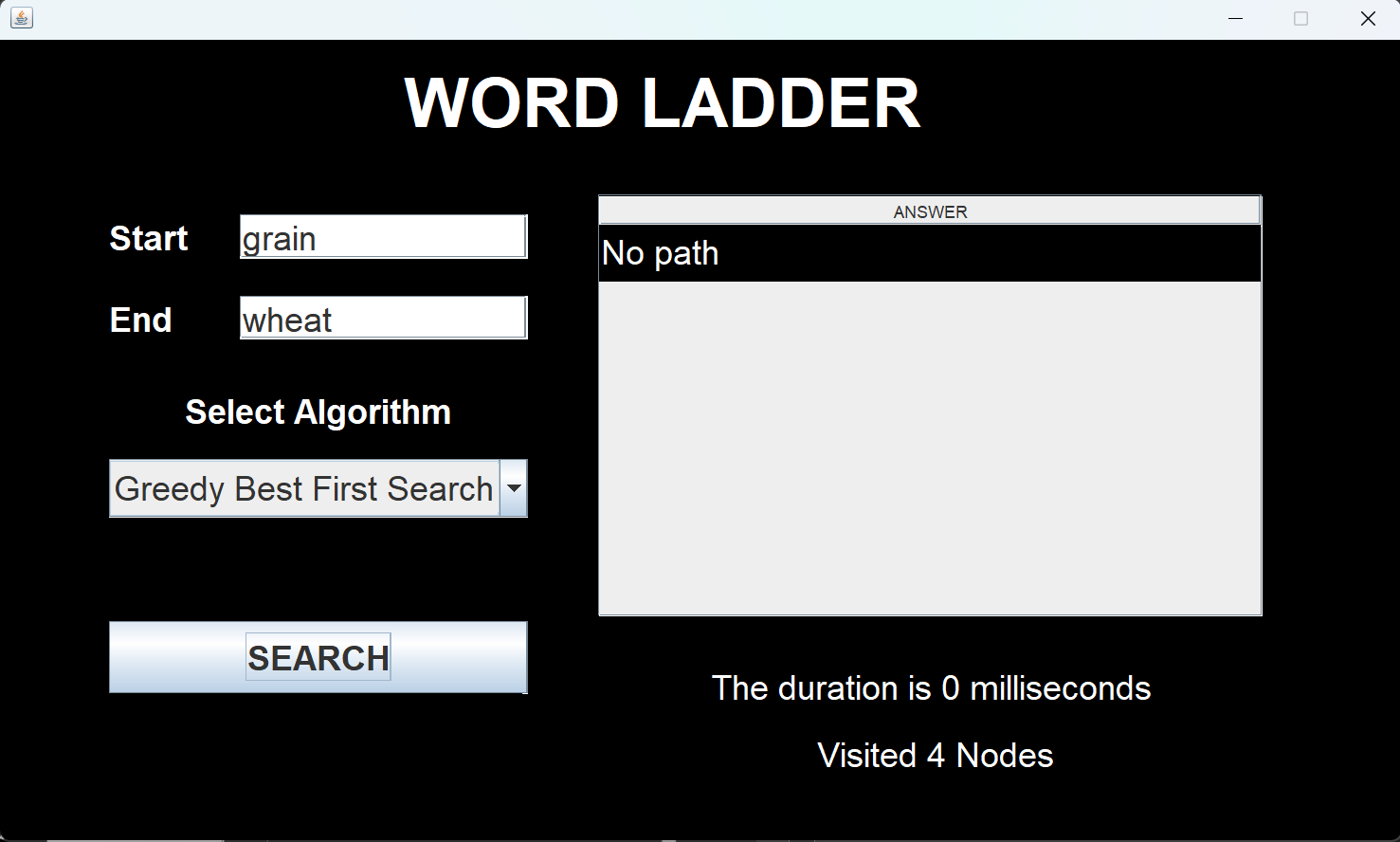


1. Grain – Wheat
2. UCS

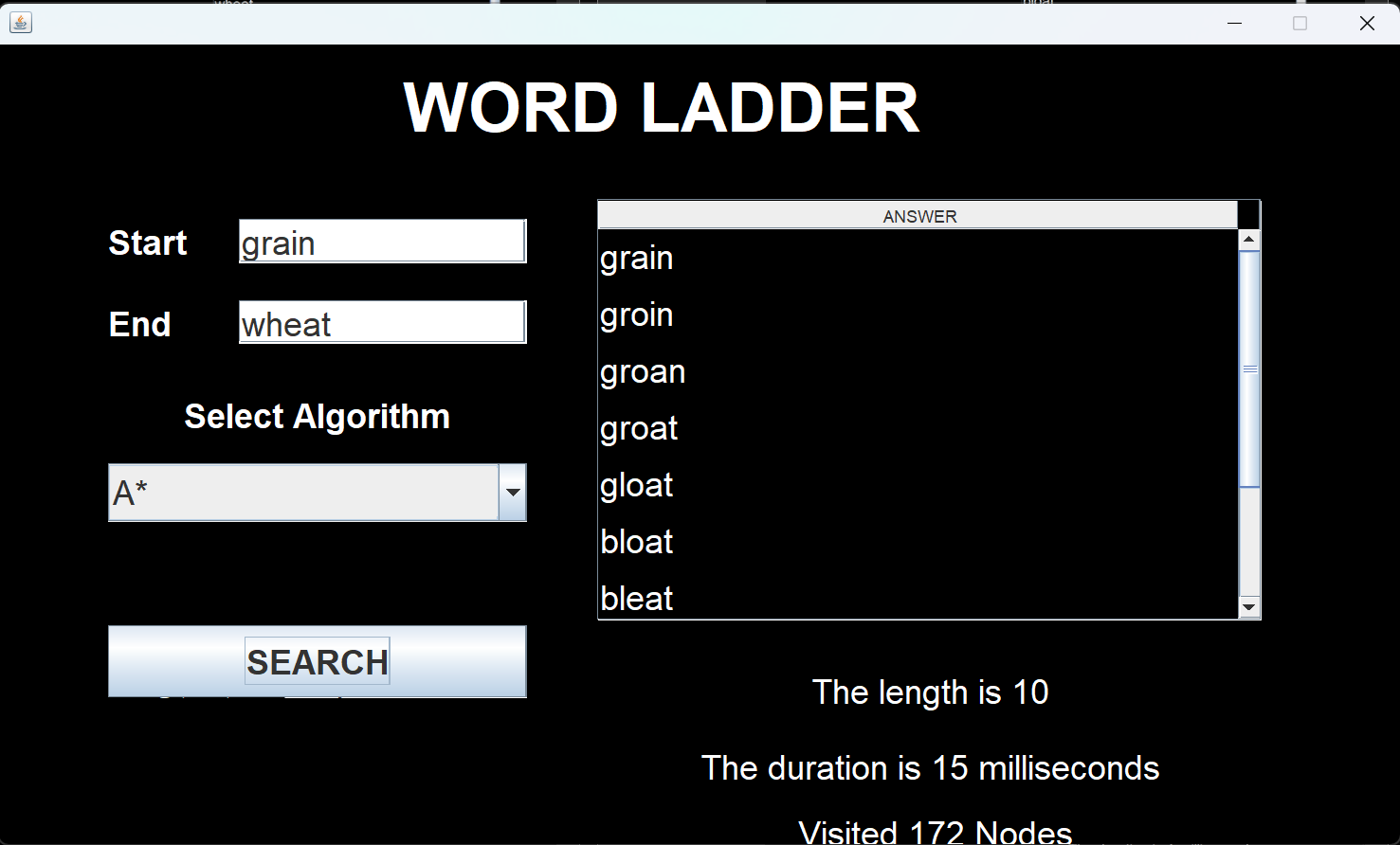


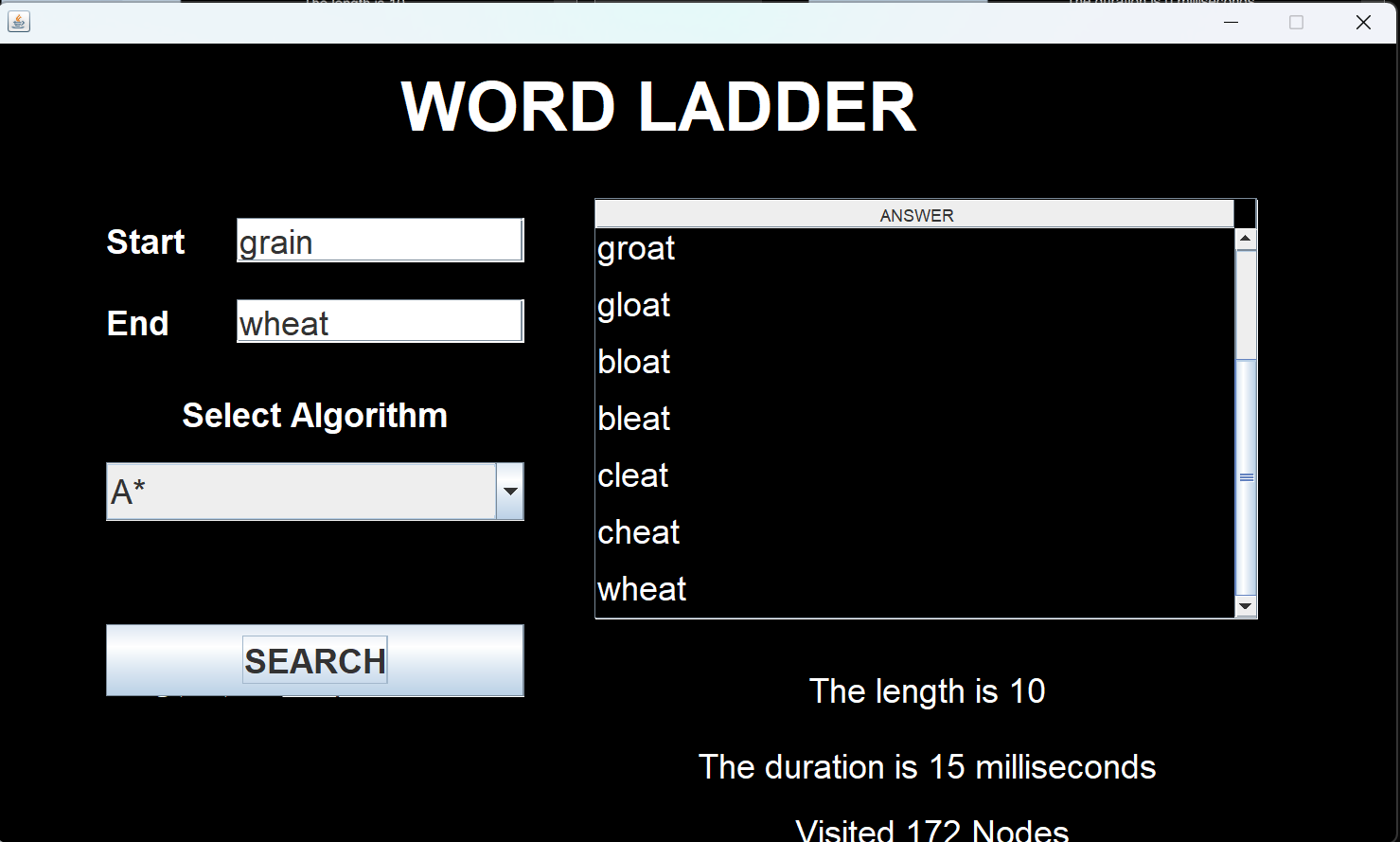


1. GBFS

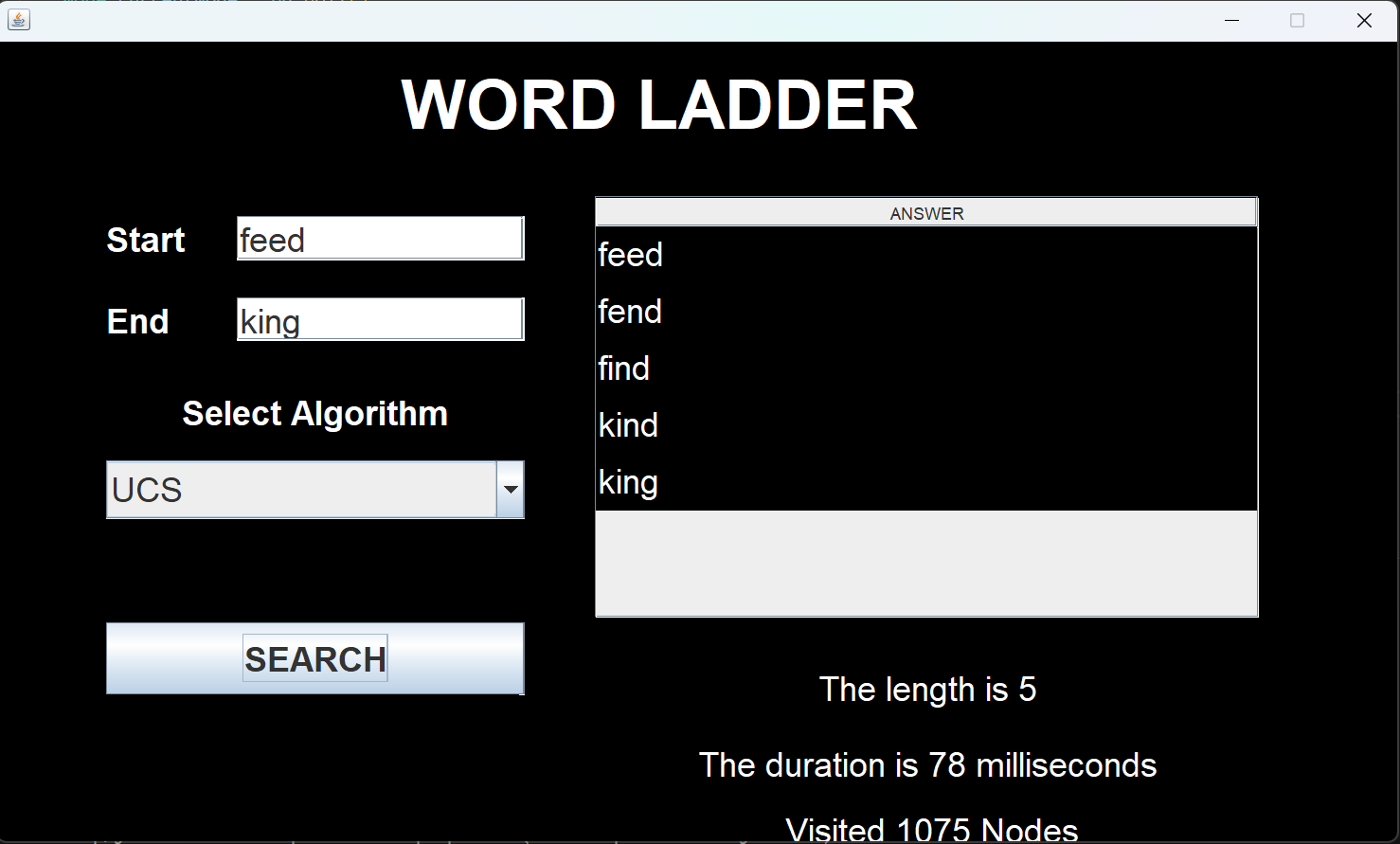


1. A Star

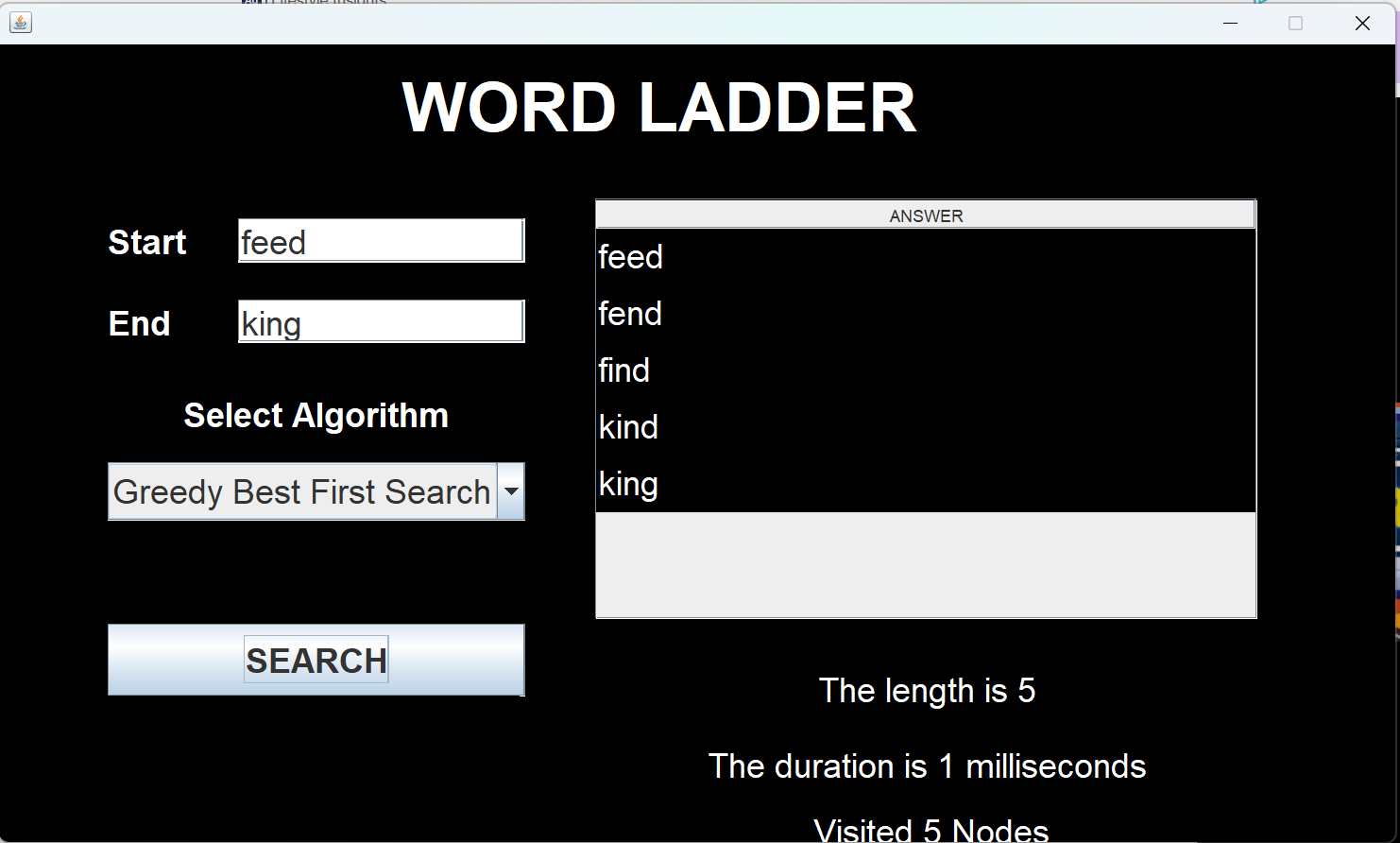




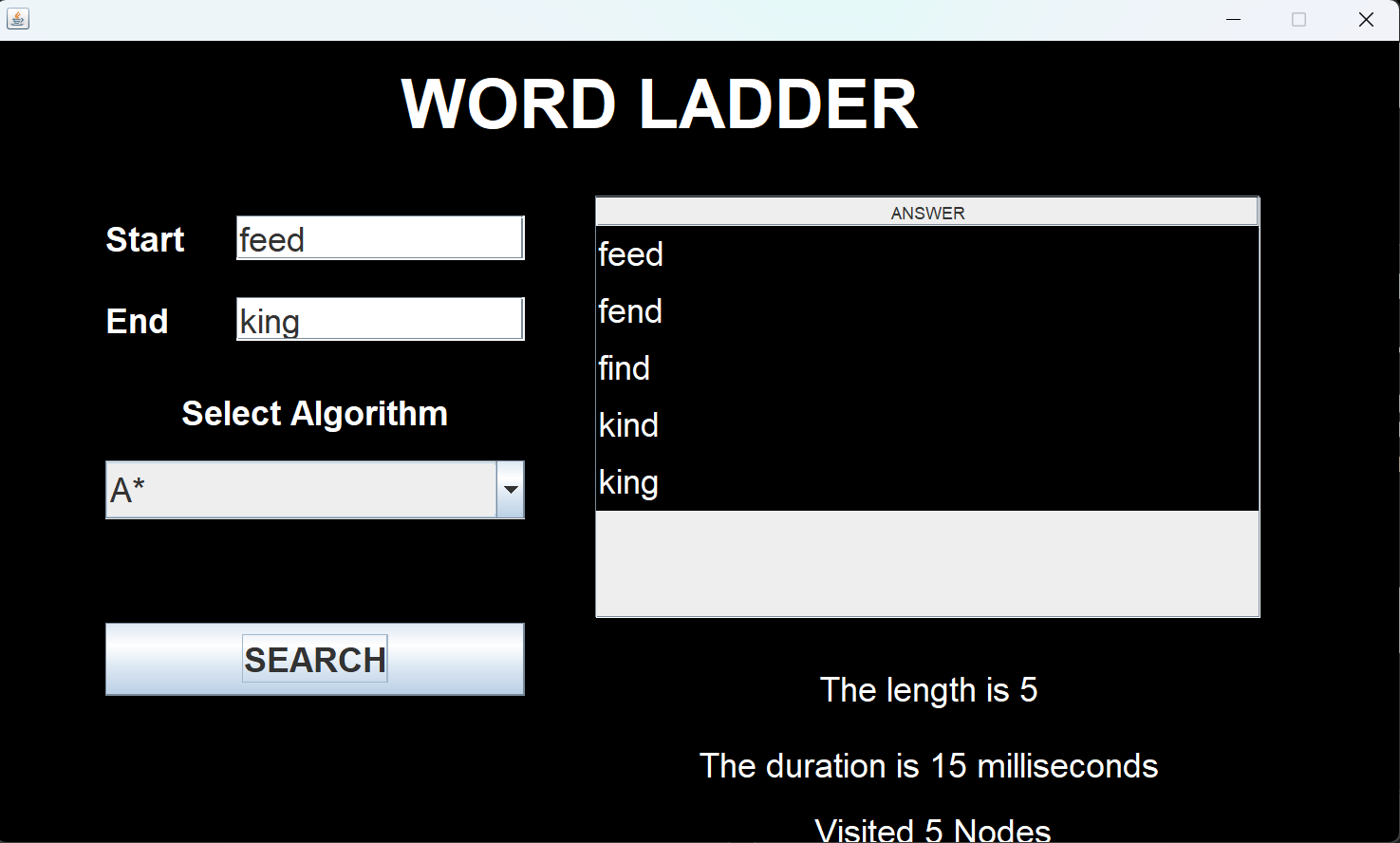
1. Feed – King
2. UCS



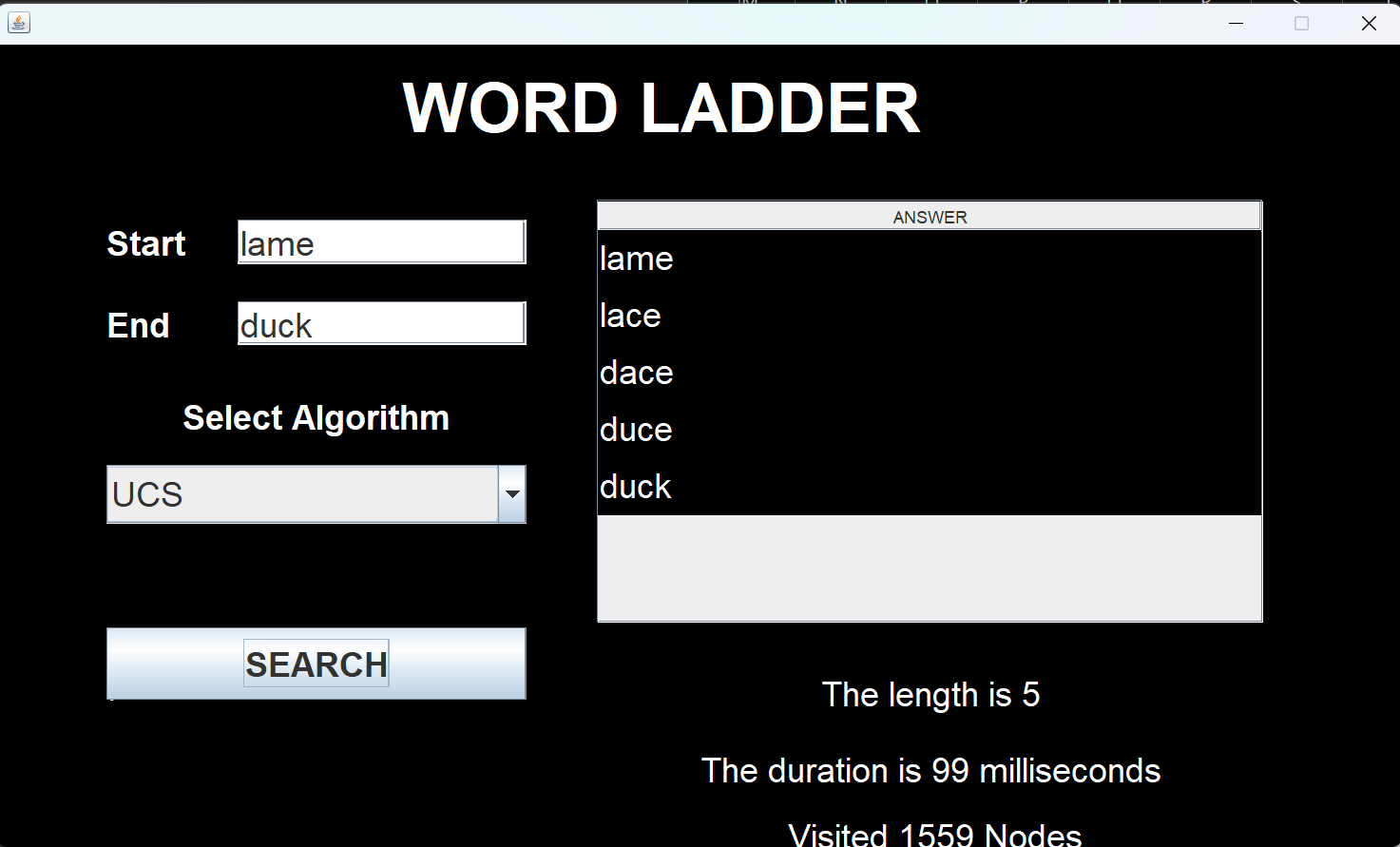
1. GBFS



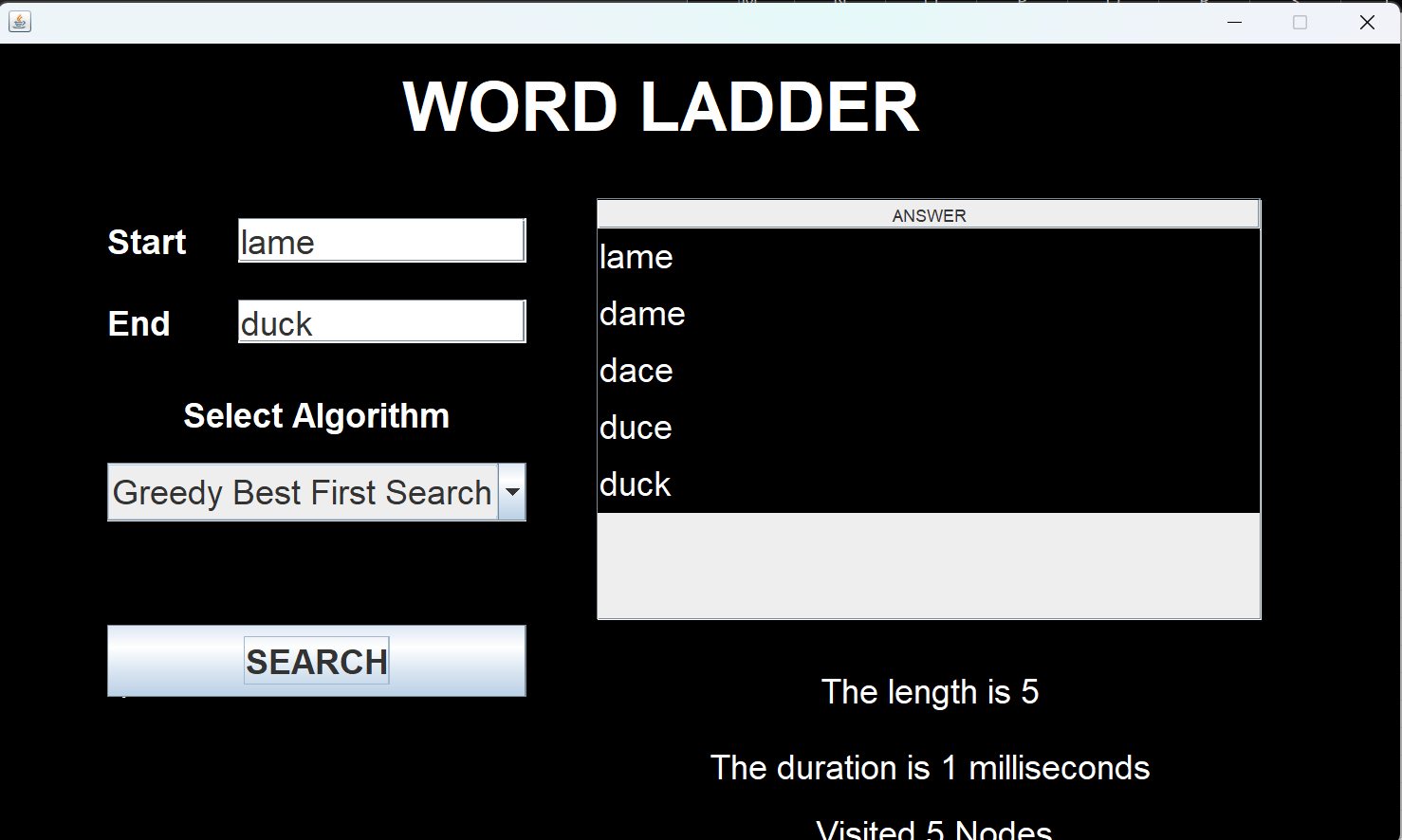
1. A Star



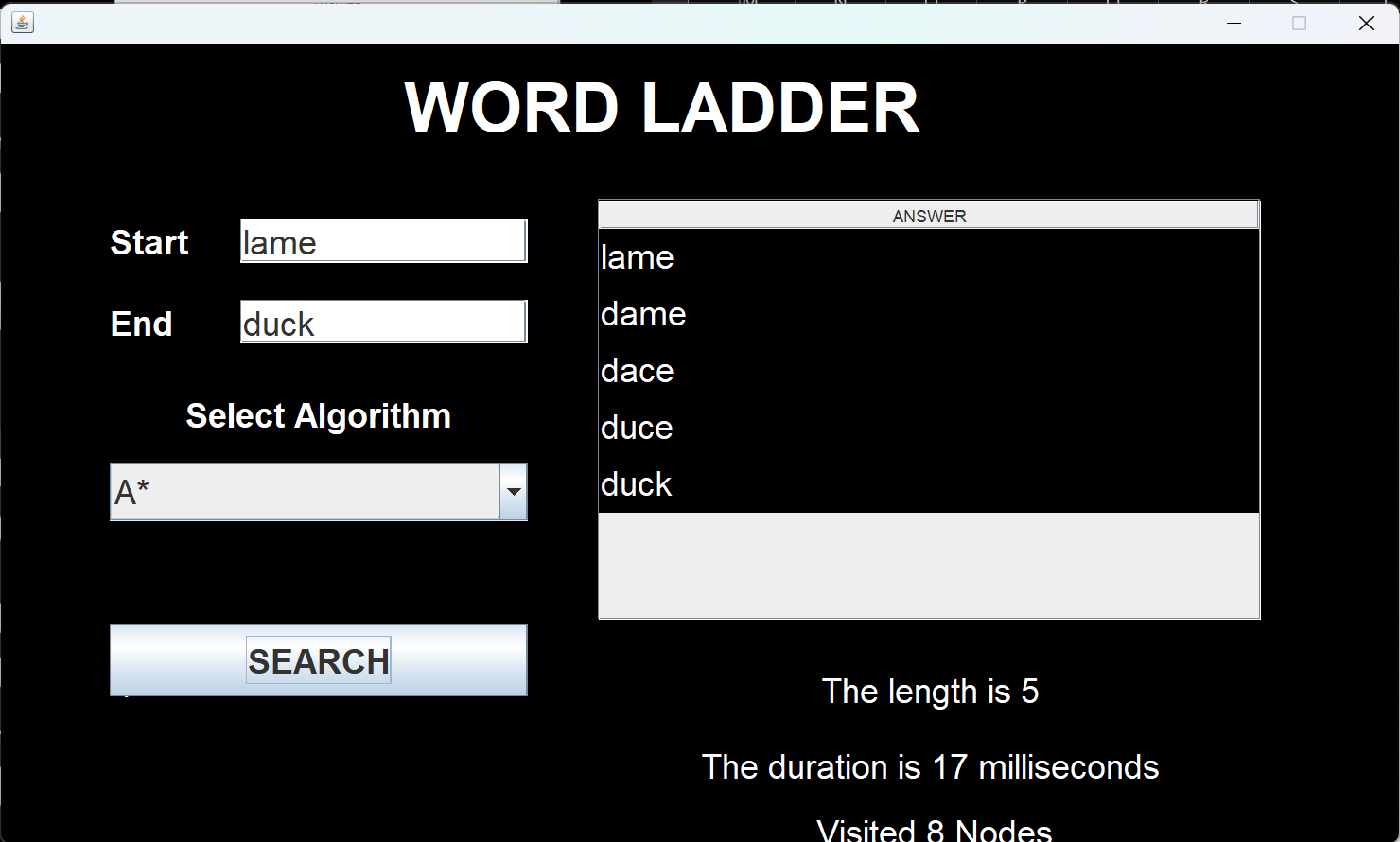
1. Lame – Duck
2. UCS



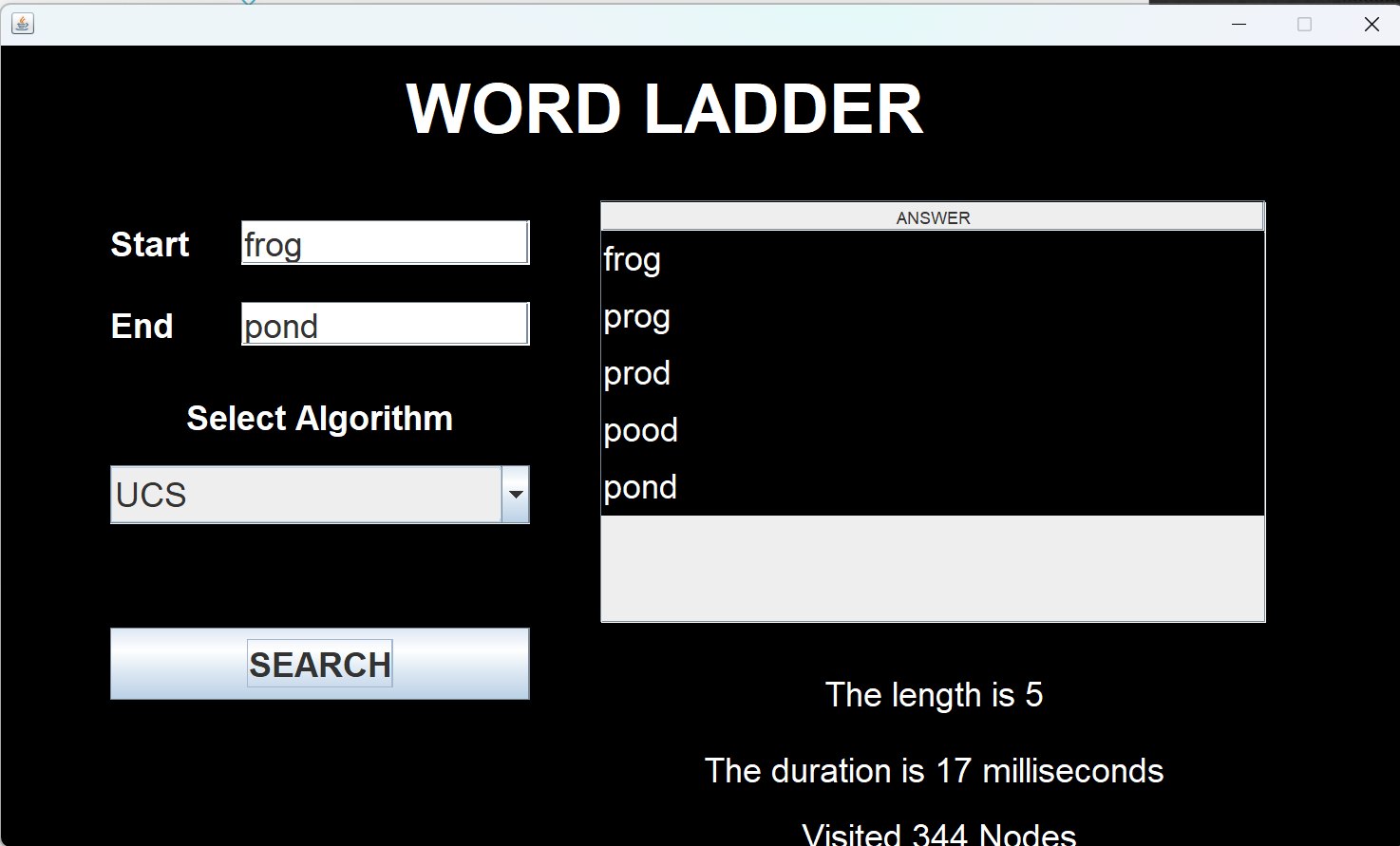
1. GBFS



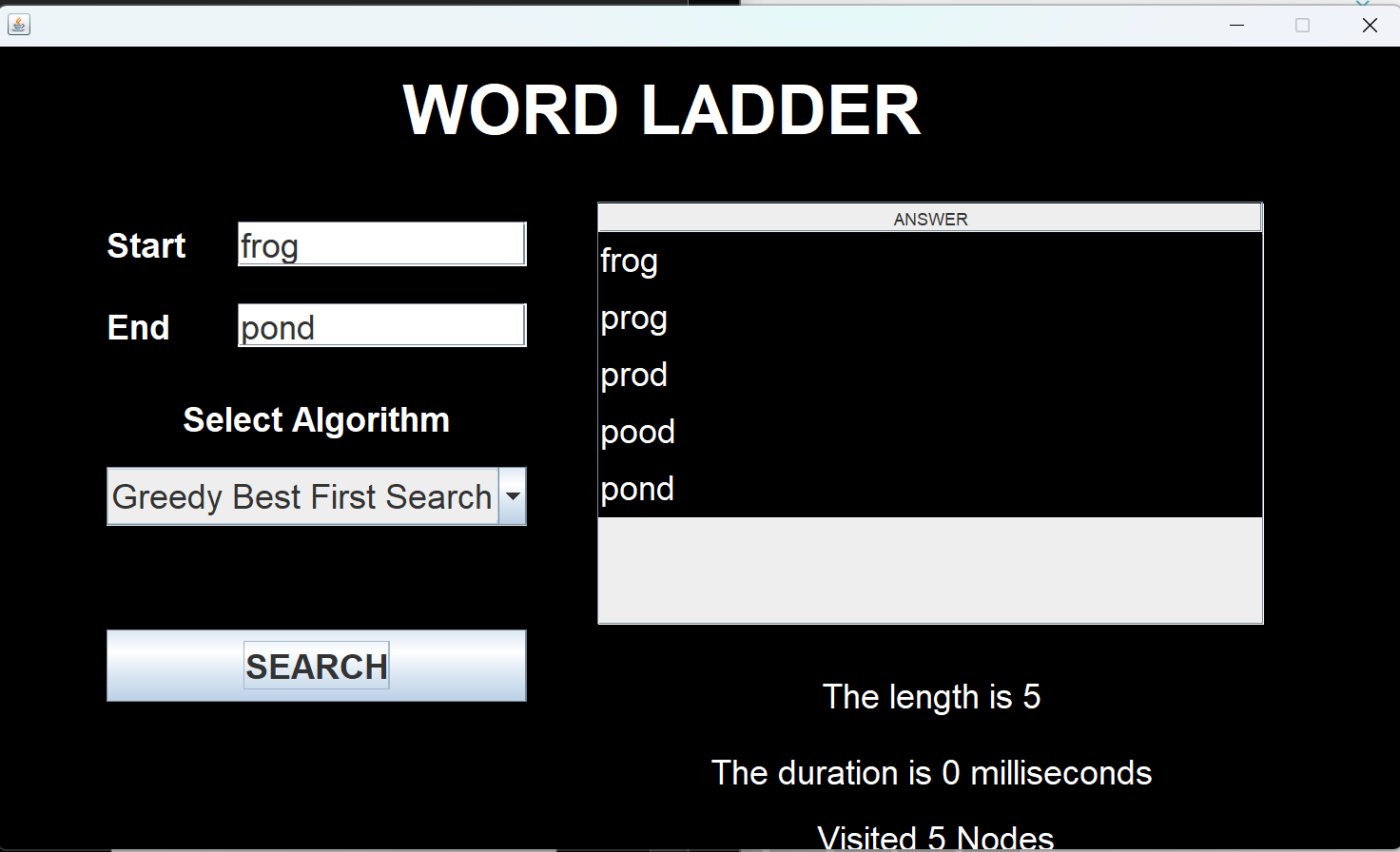
1. A Star



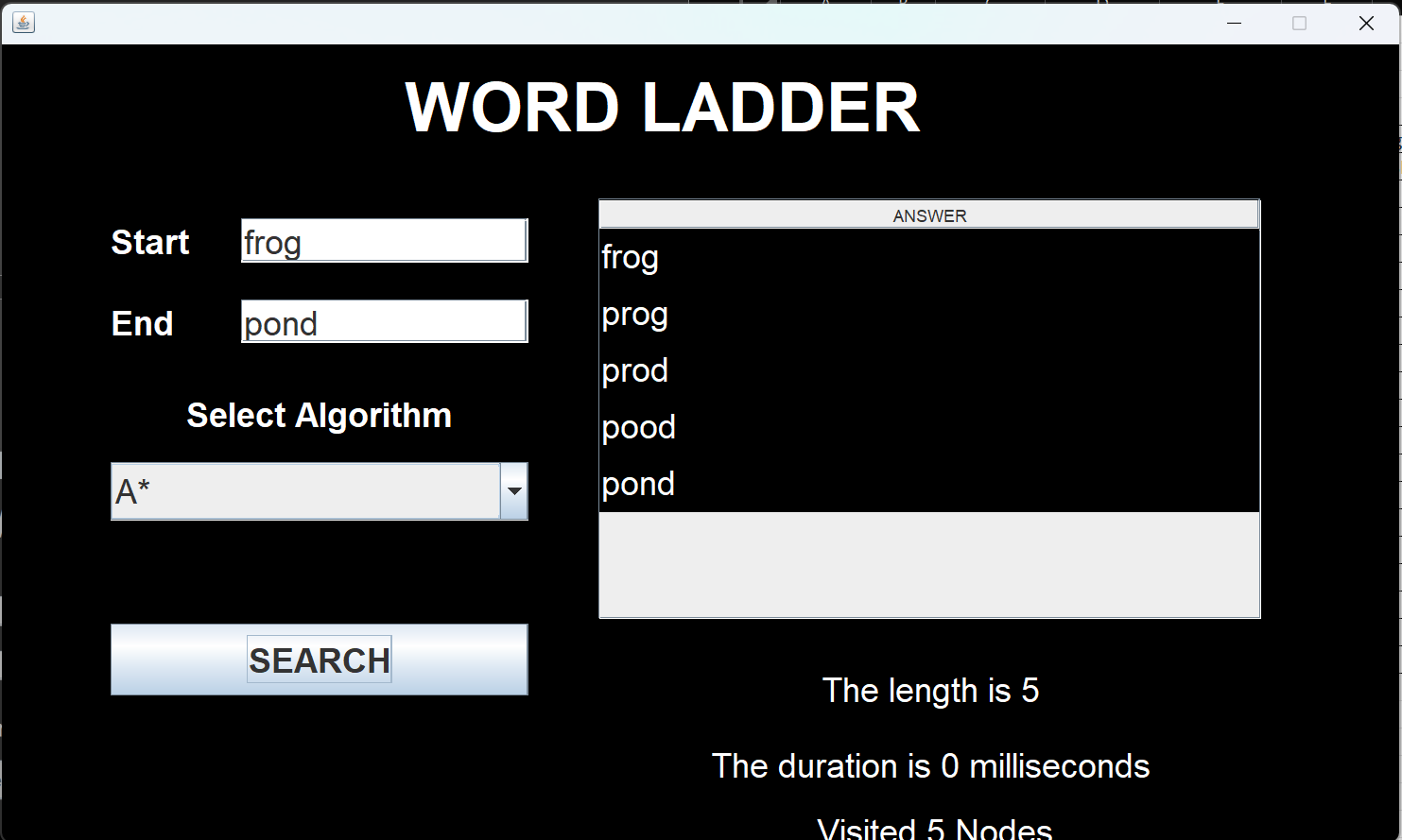
1. Frog – Pond
2. UCS



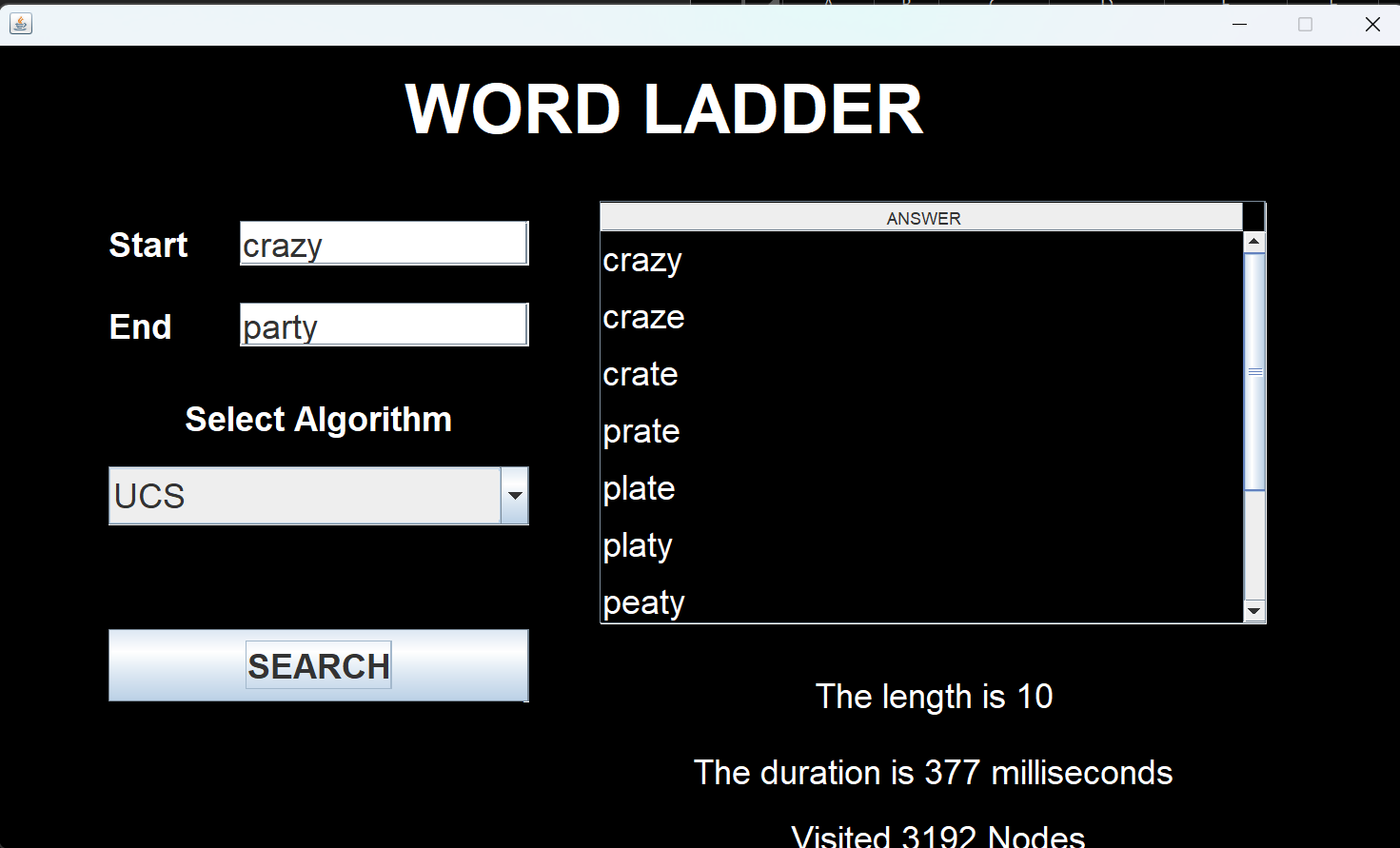
1. GBFS

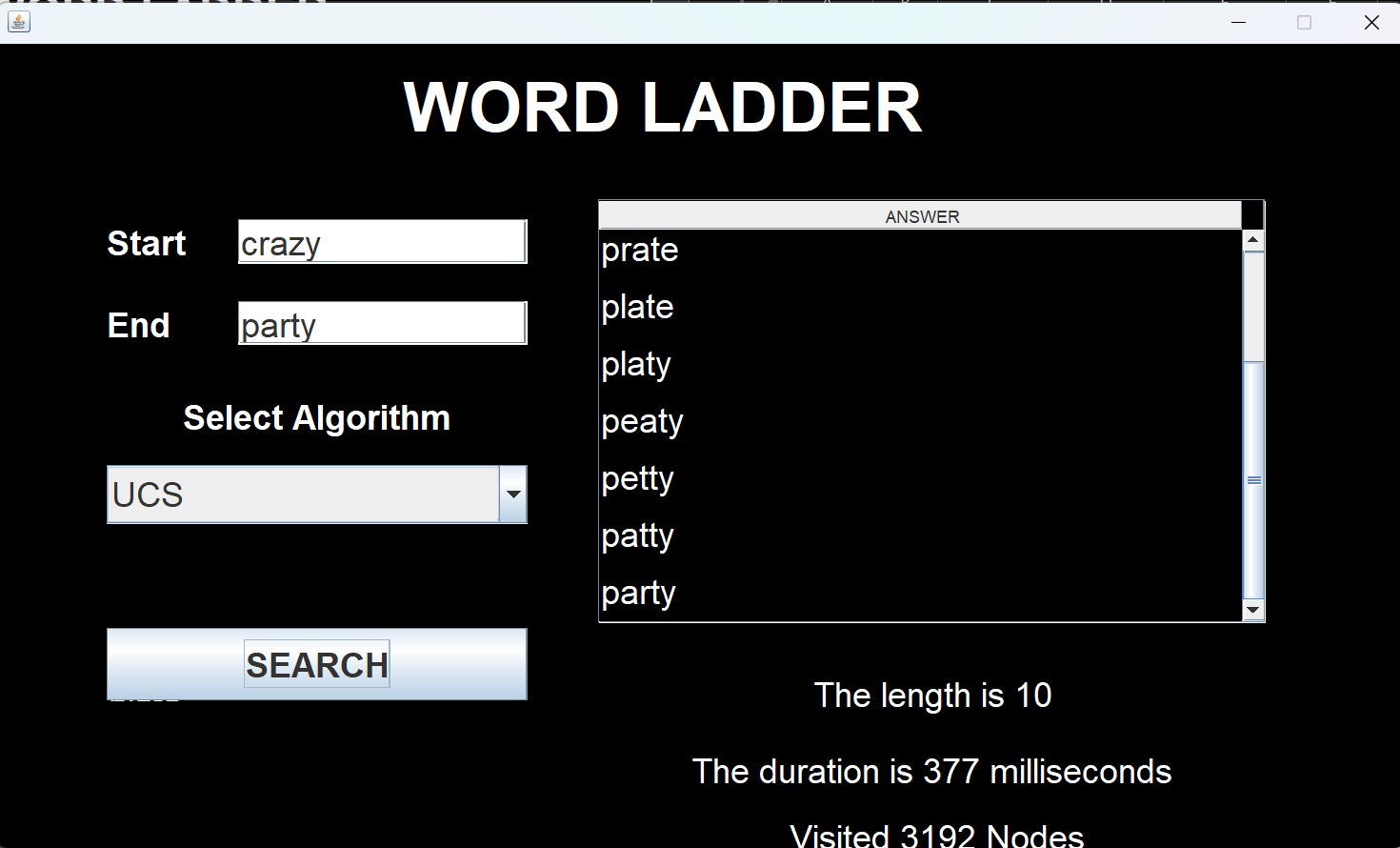


1. A Star

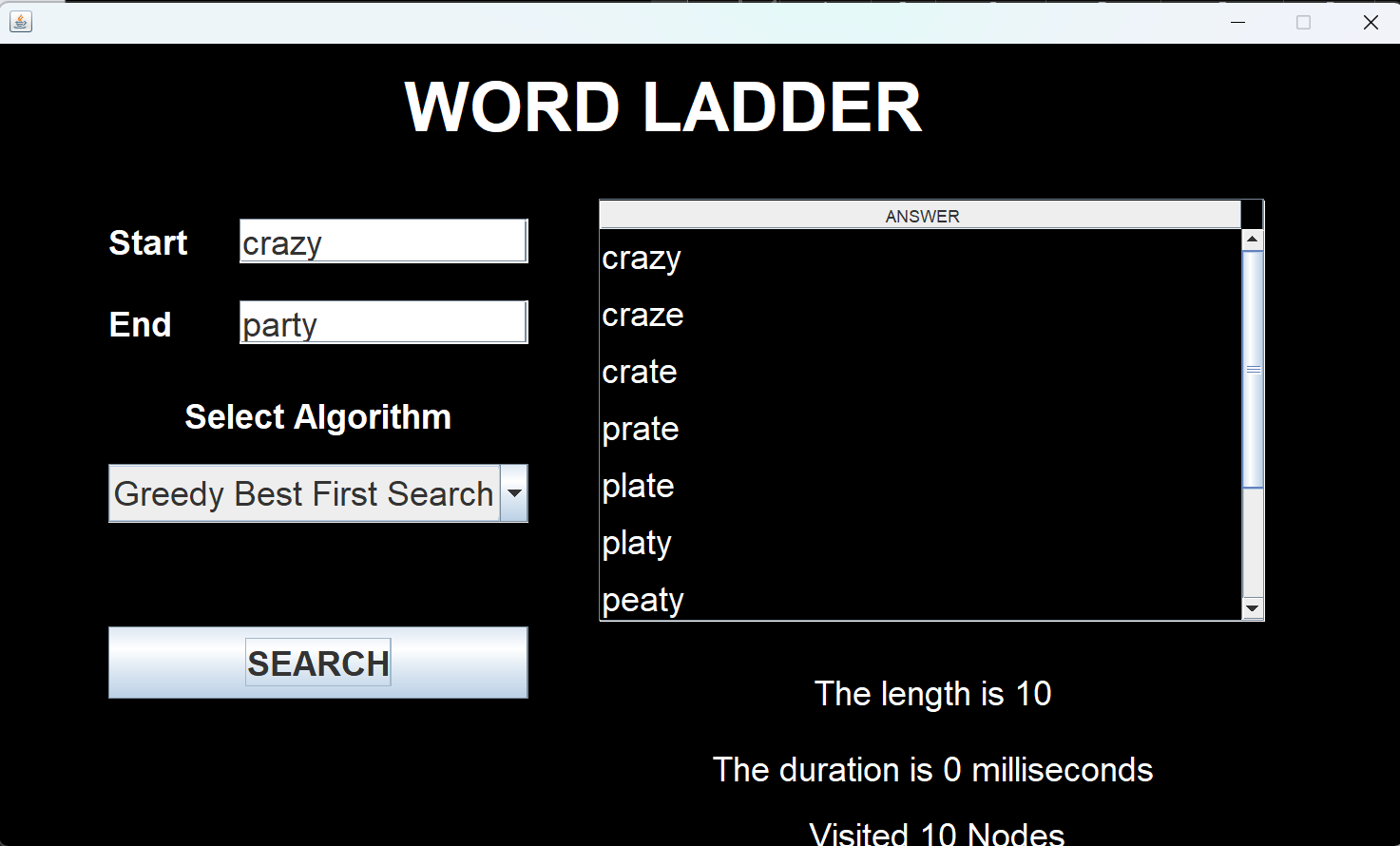


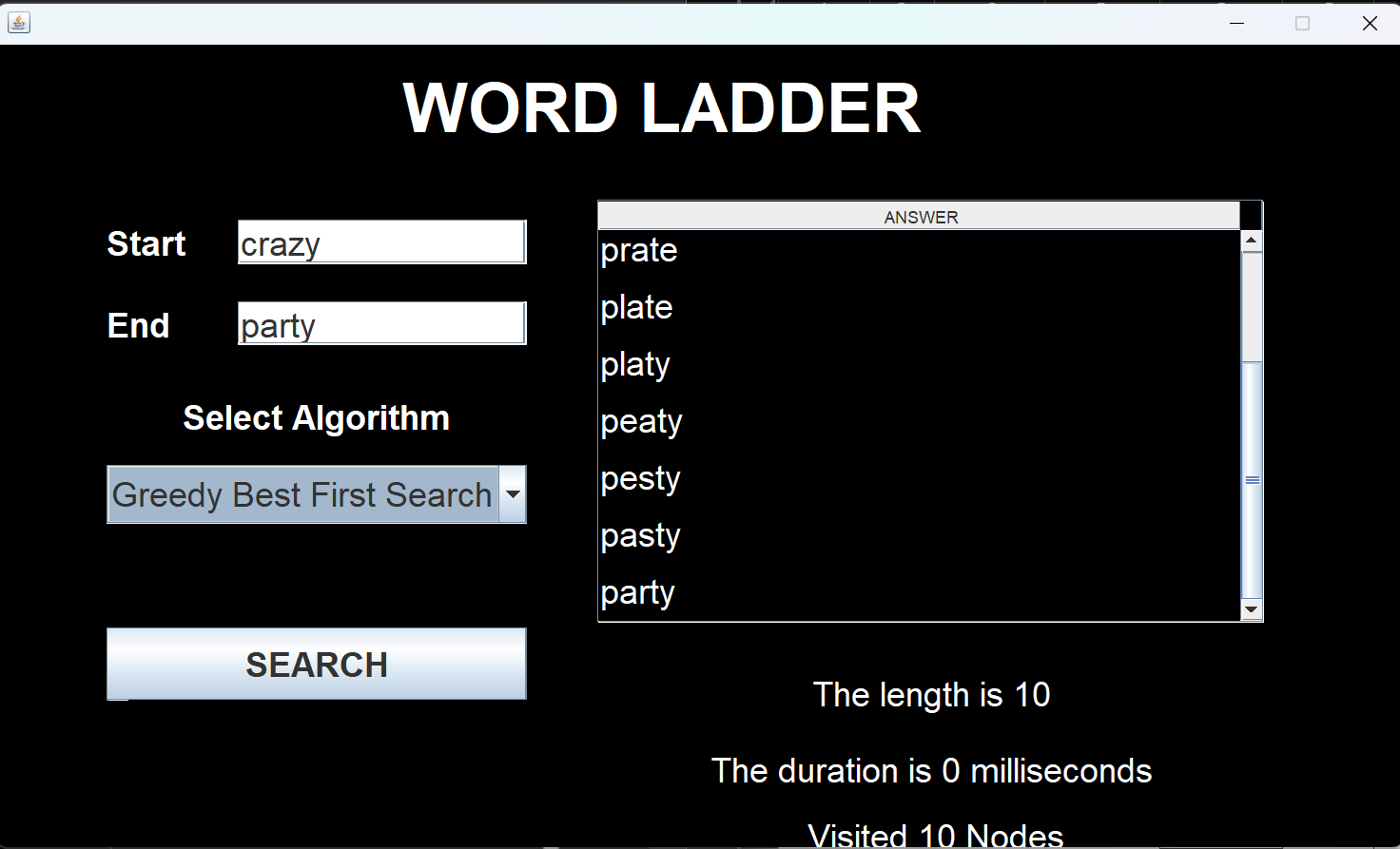
1. Crazy – Party
2. UCS



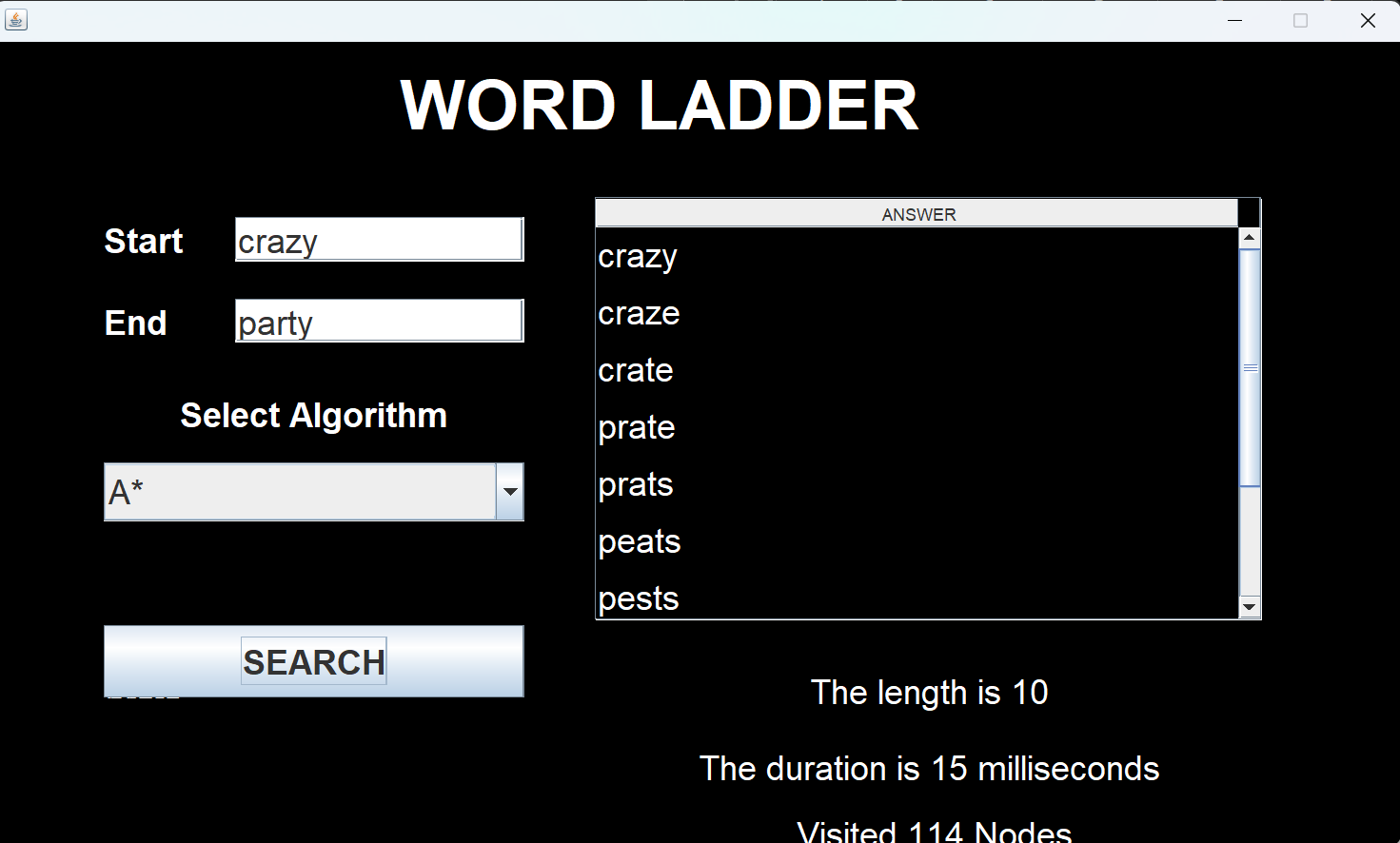


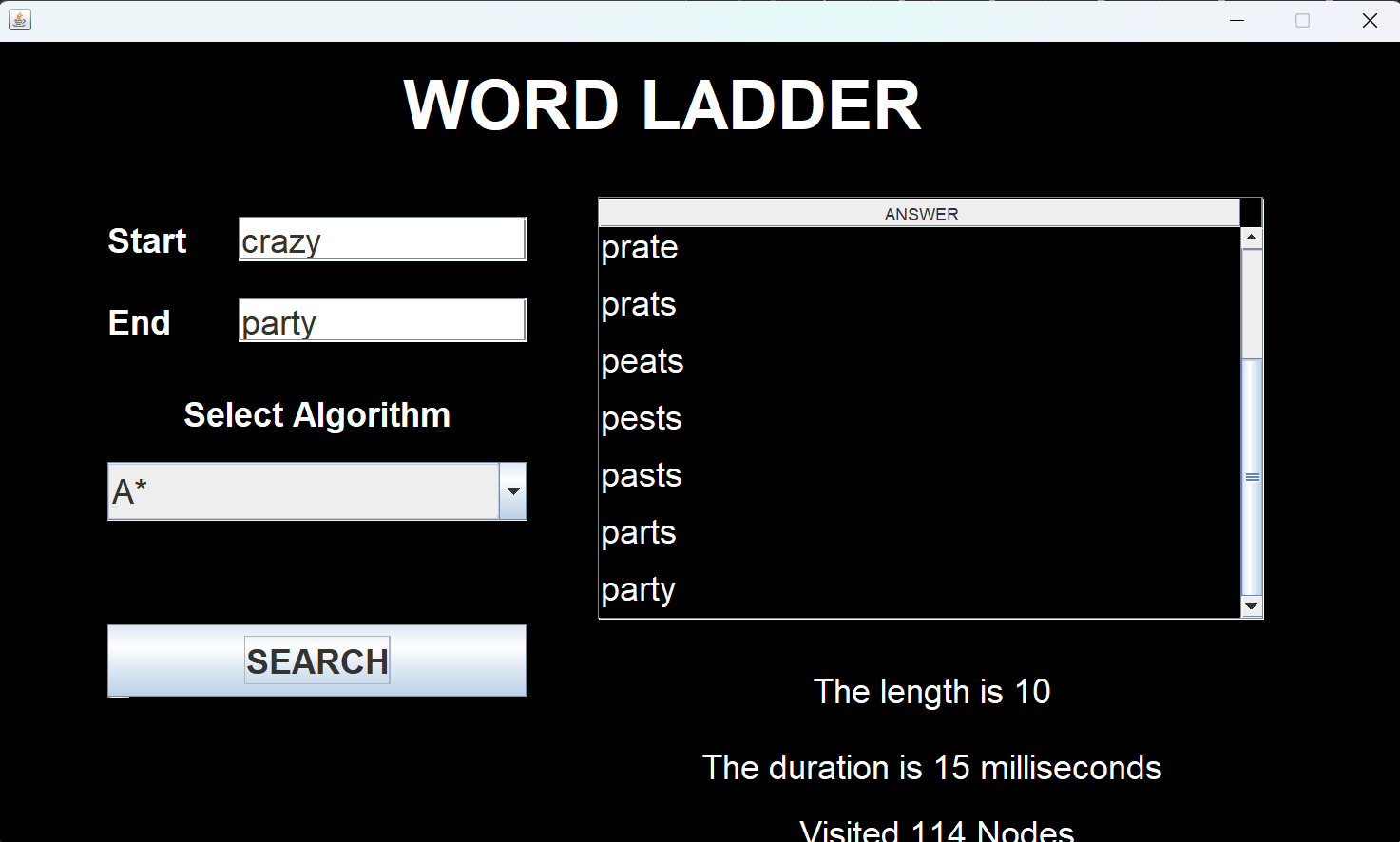
1. GBFS



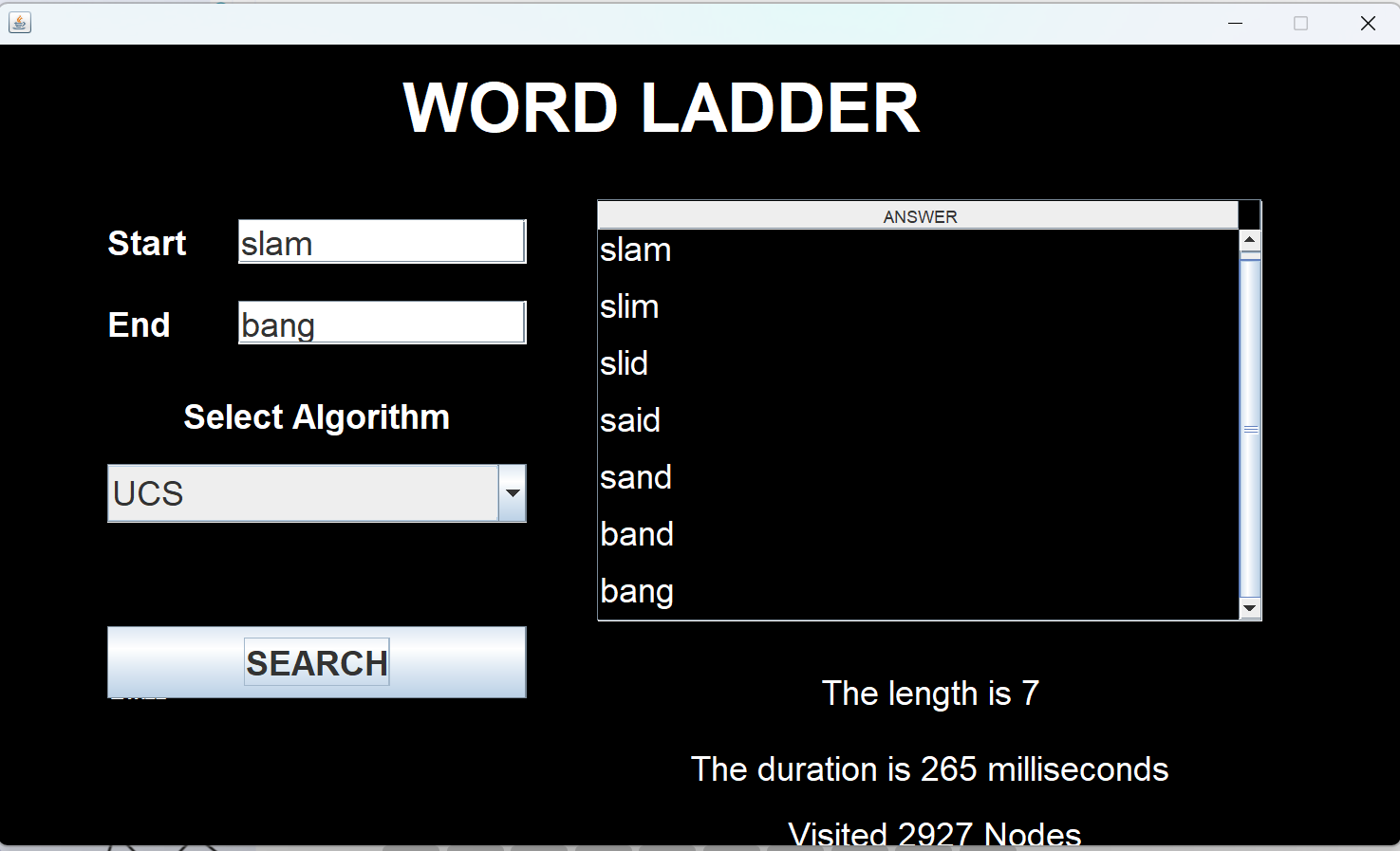


1. A Star

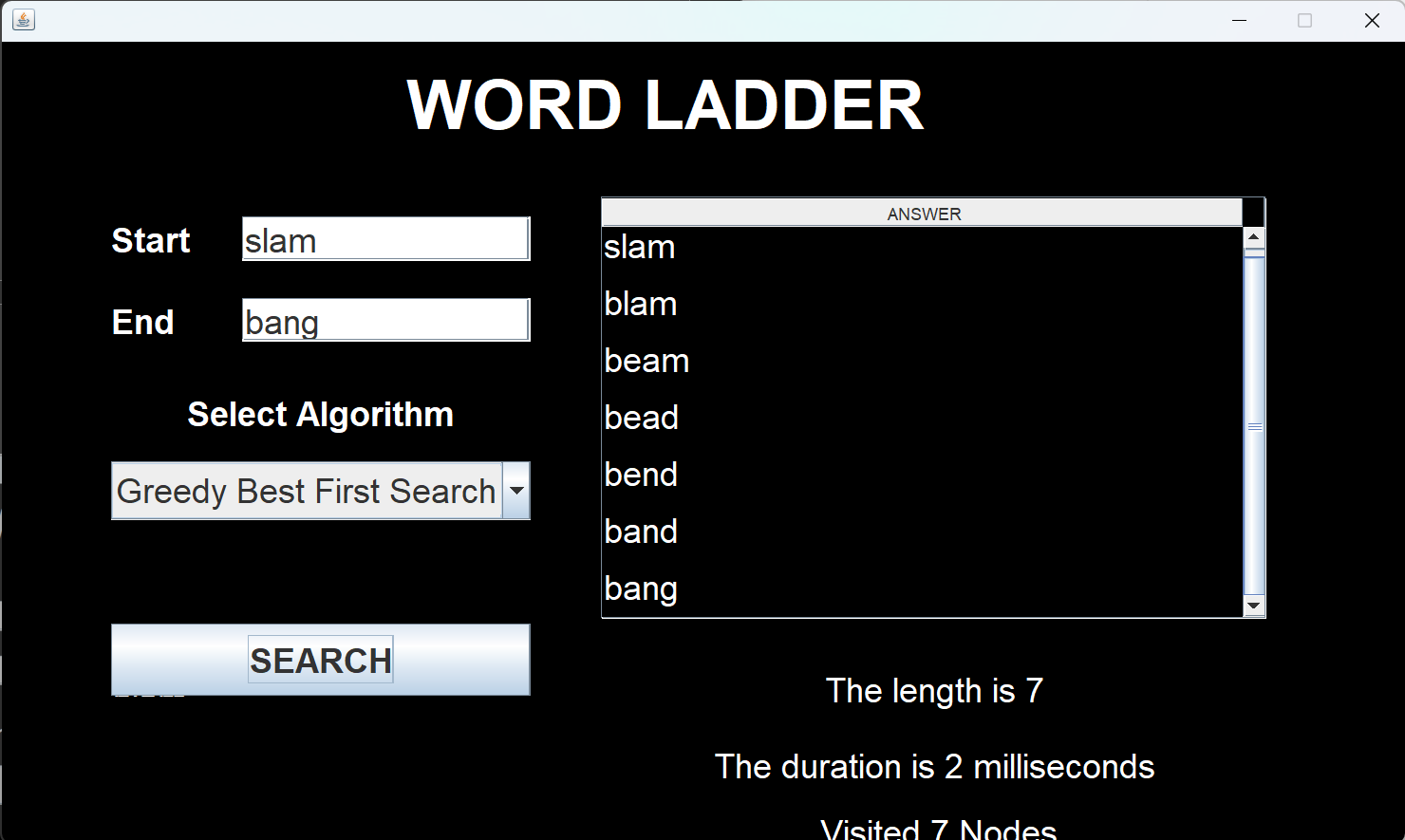




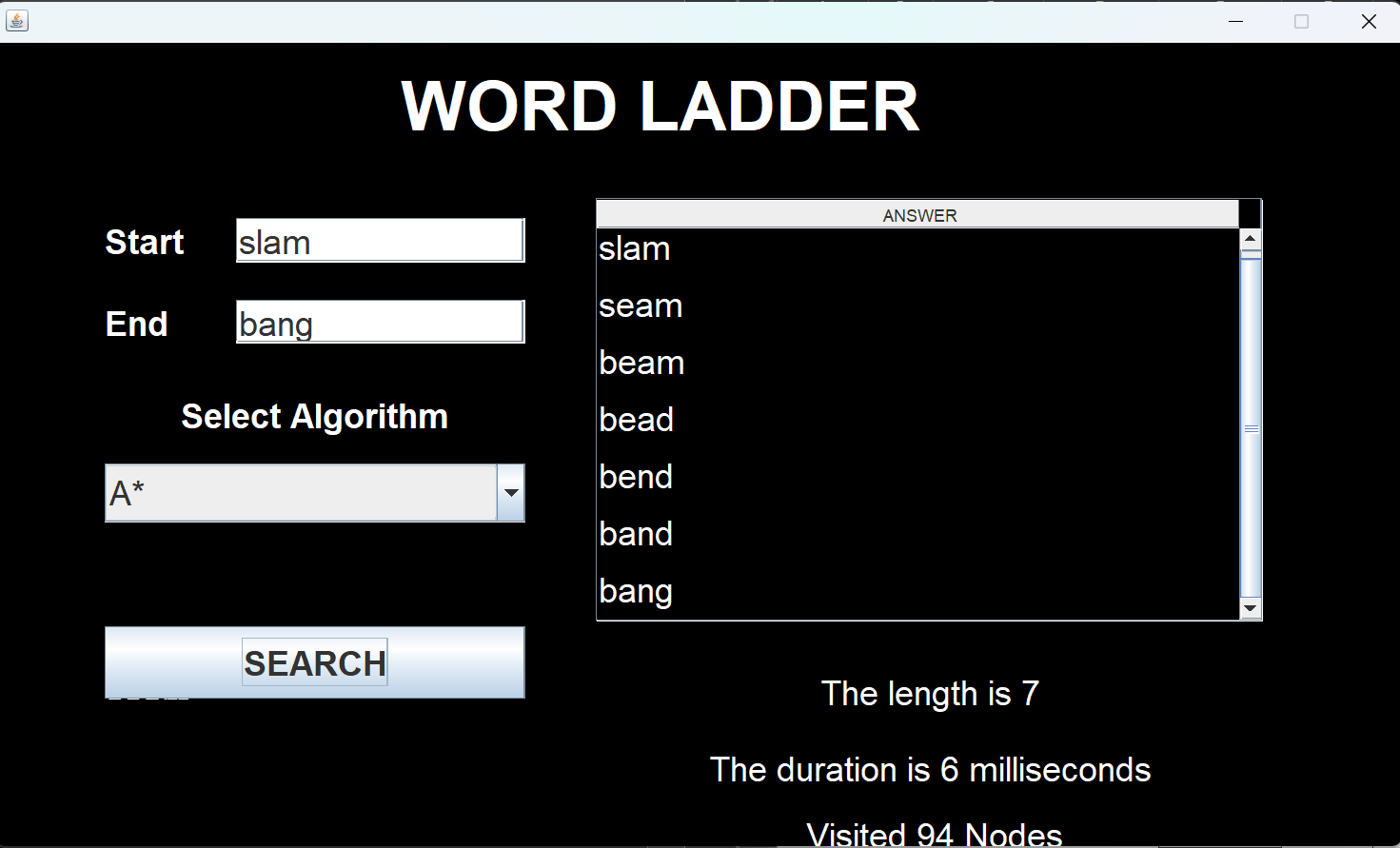
1. Slam – Bang
2. UCS



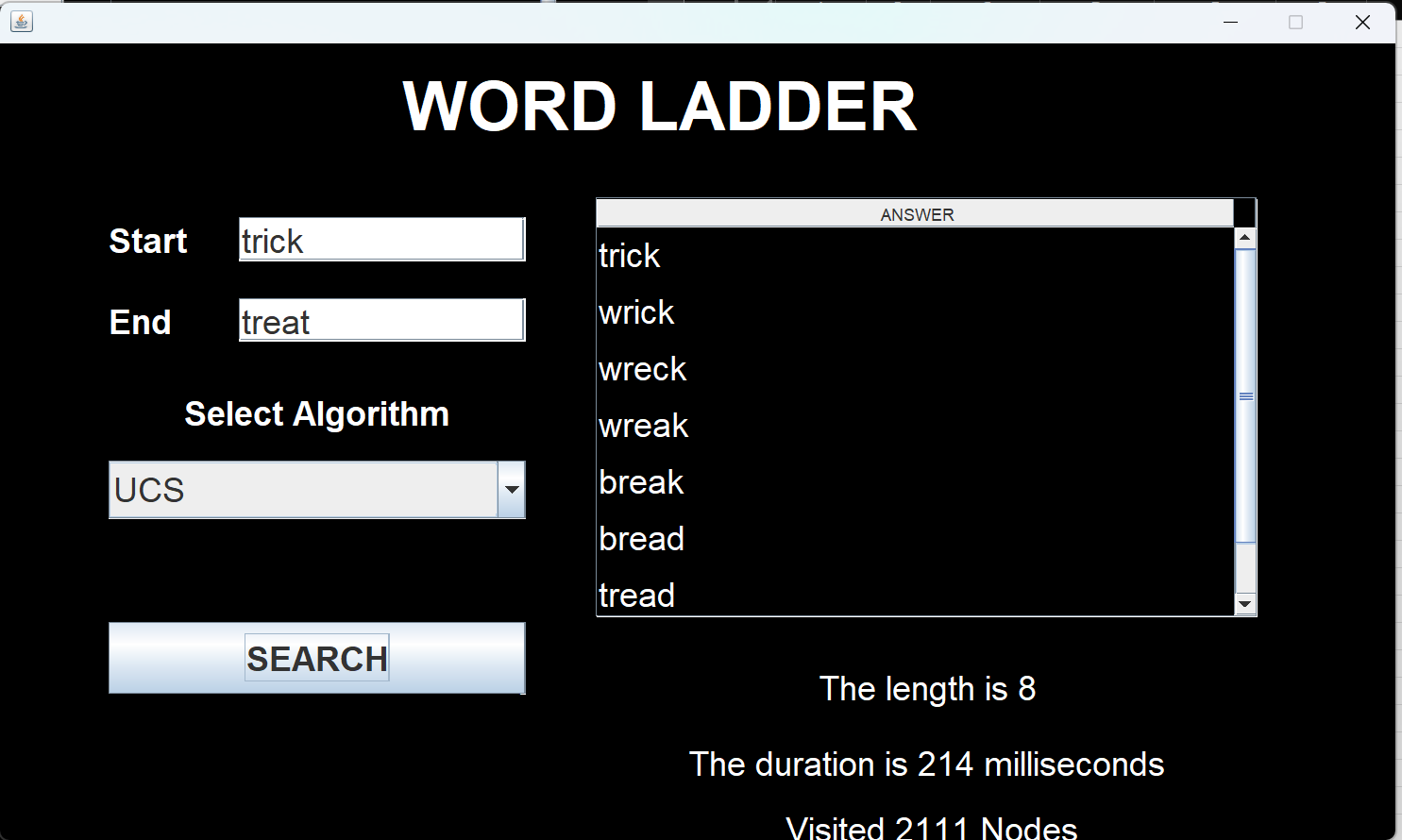
1. GBFS

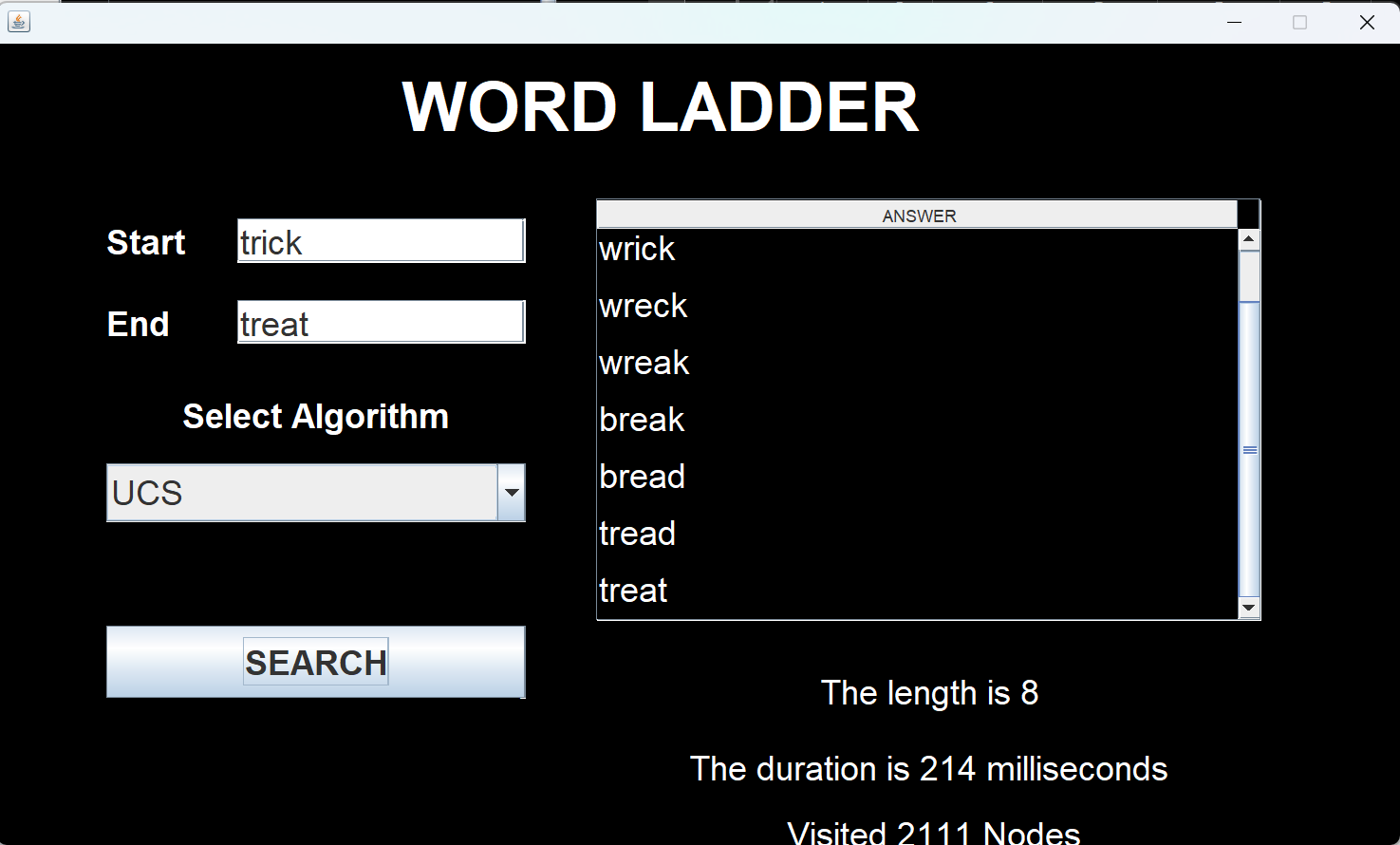


1. A Star

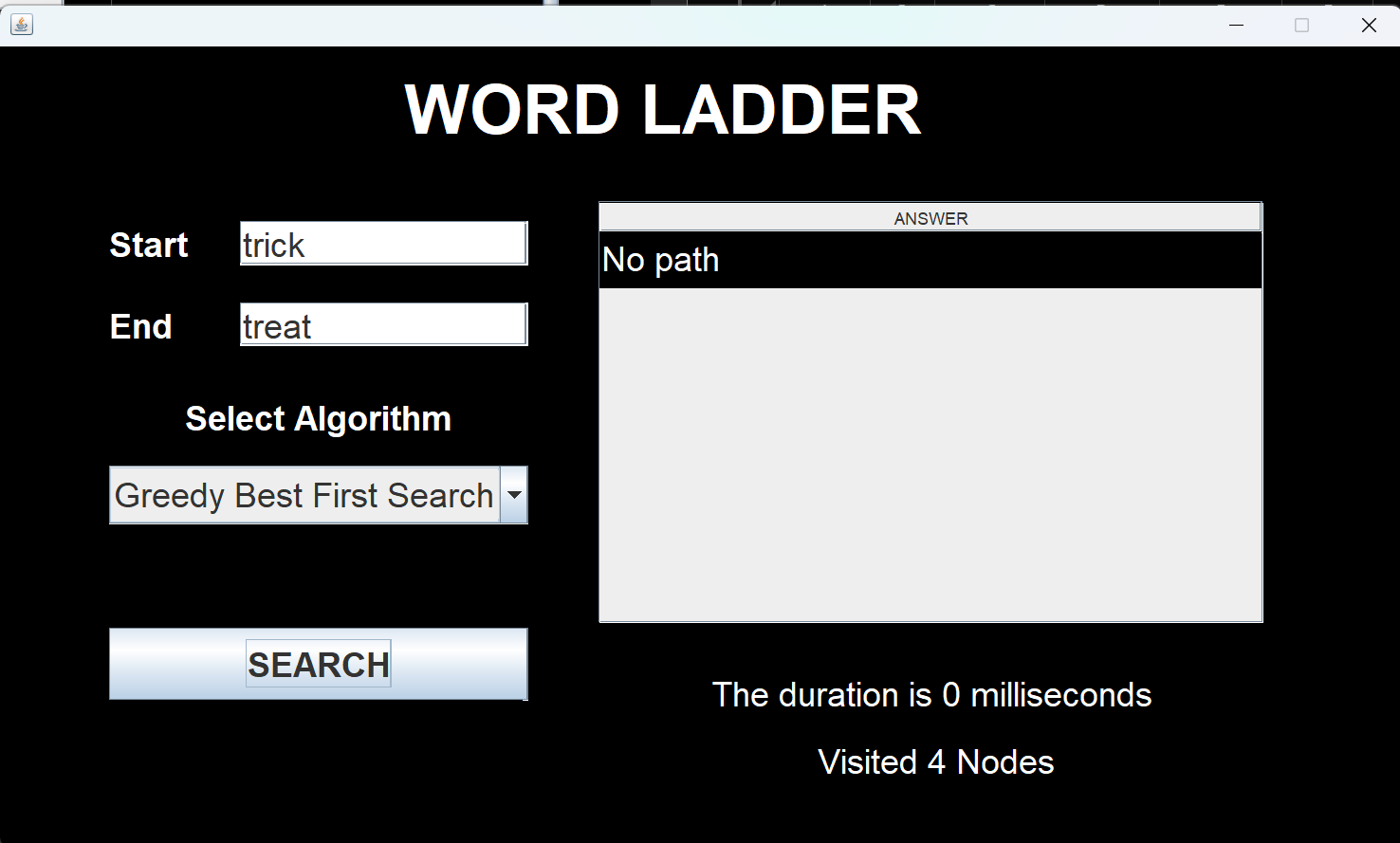


1. Trick – Treat
2. UCS

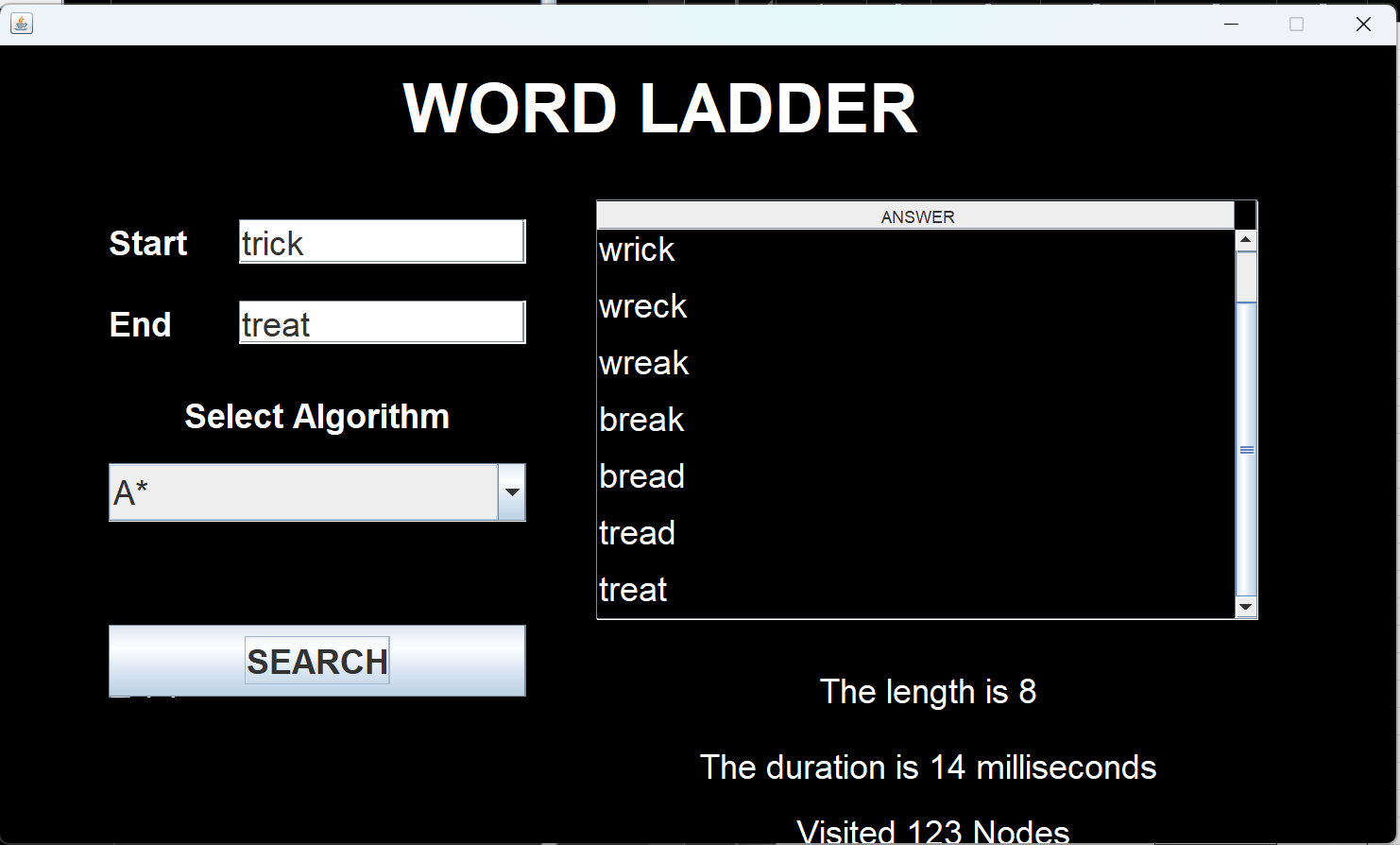


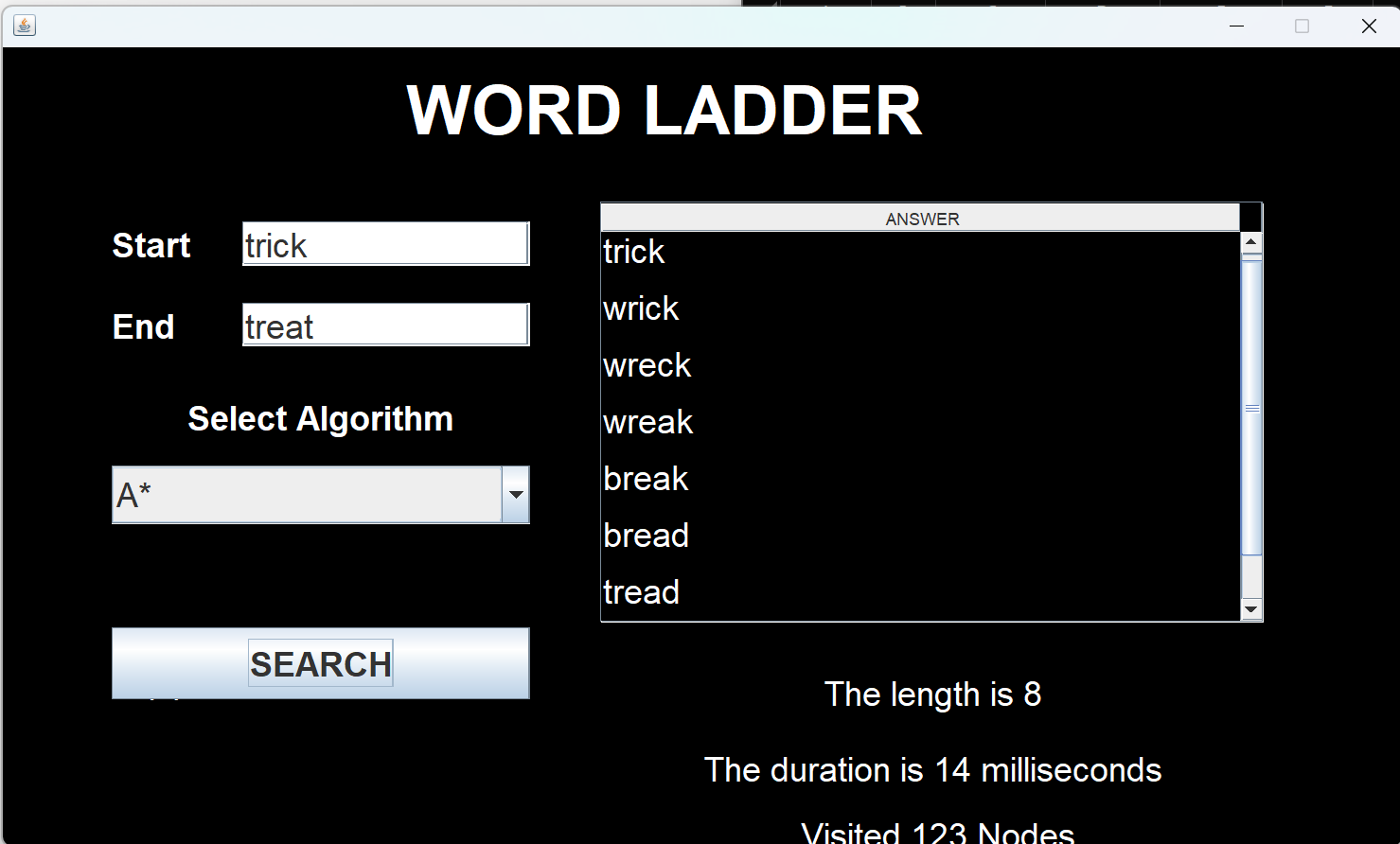


1. GBFS

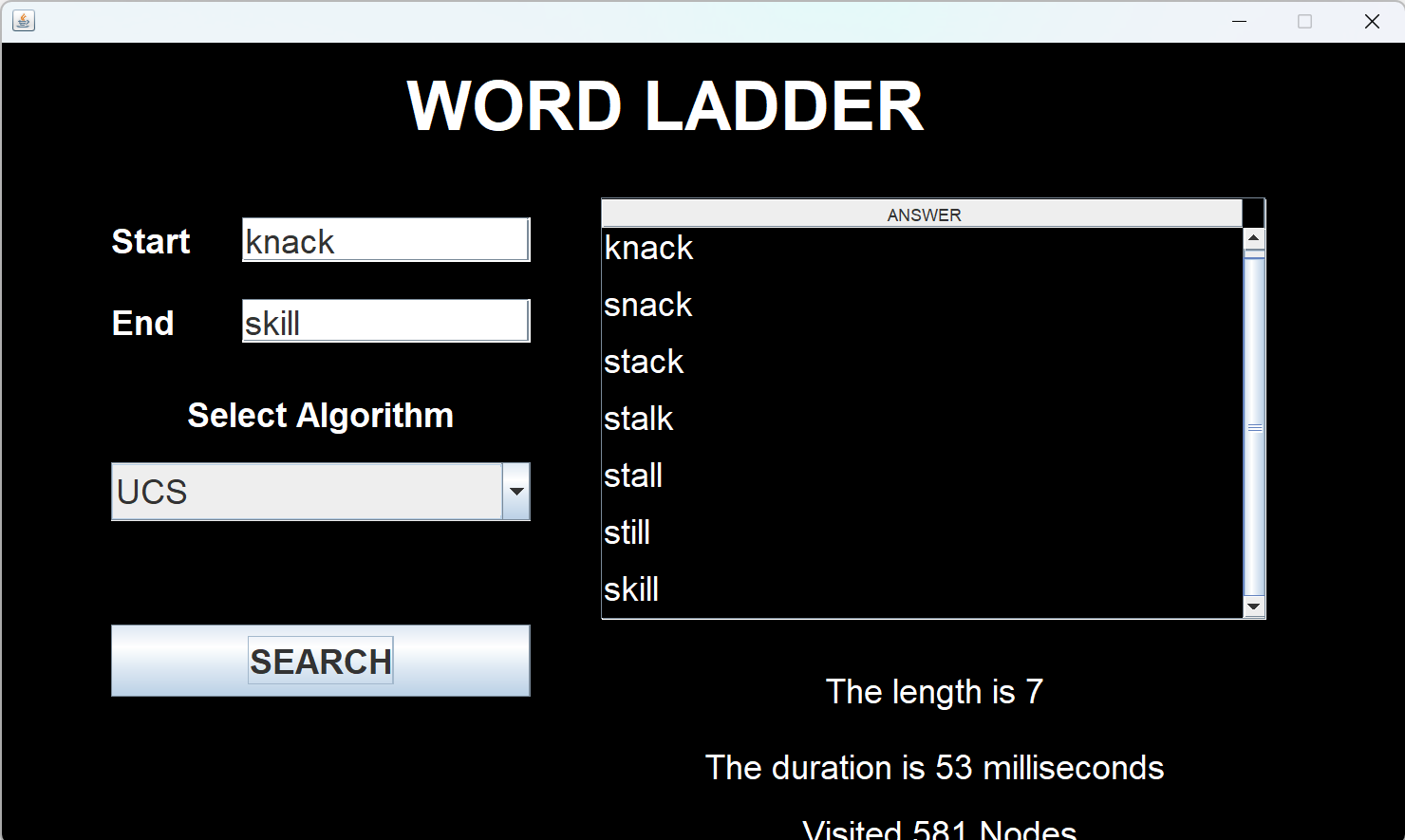


1. A Star

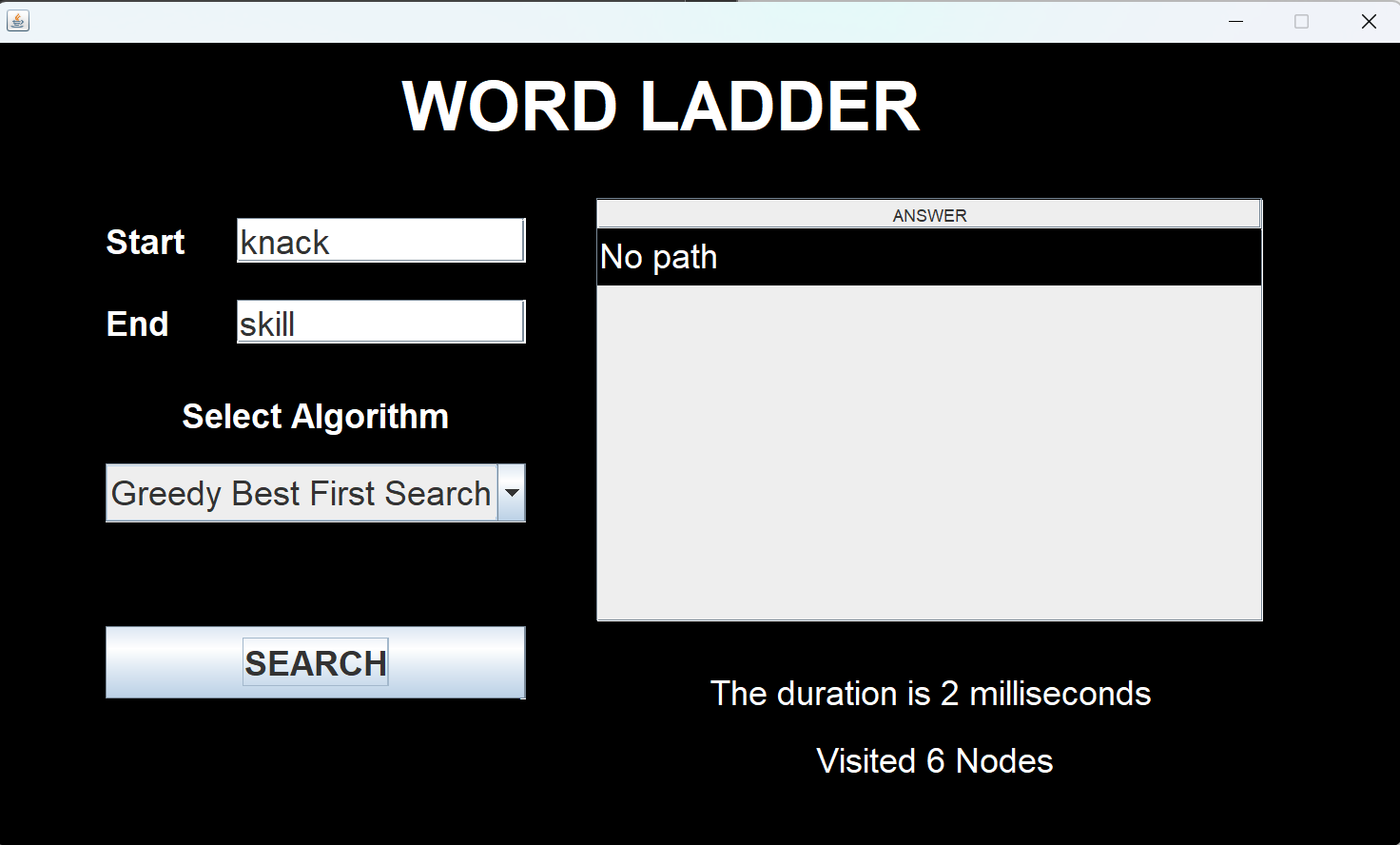




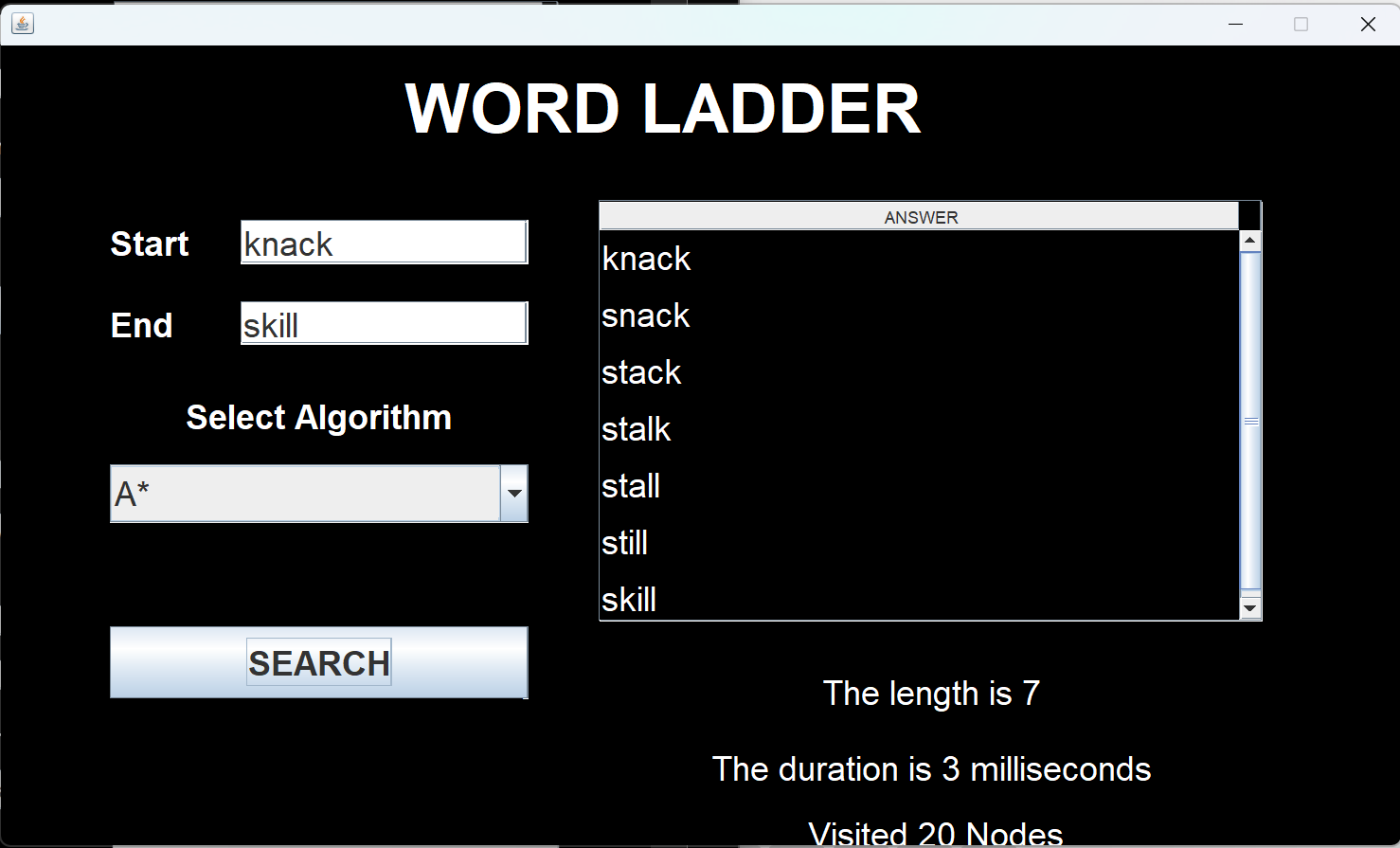
1. Knack – Skill
2. UCS



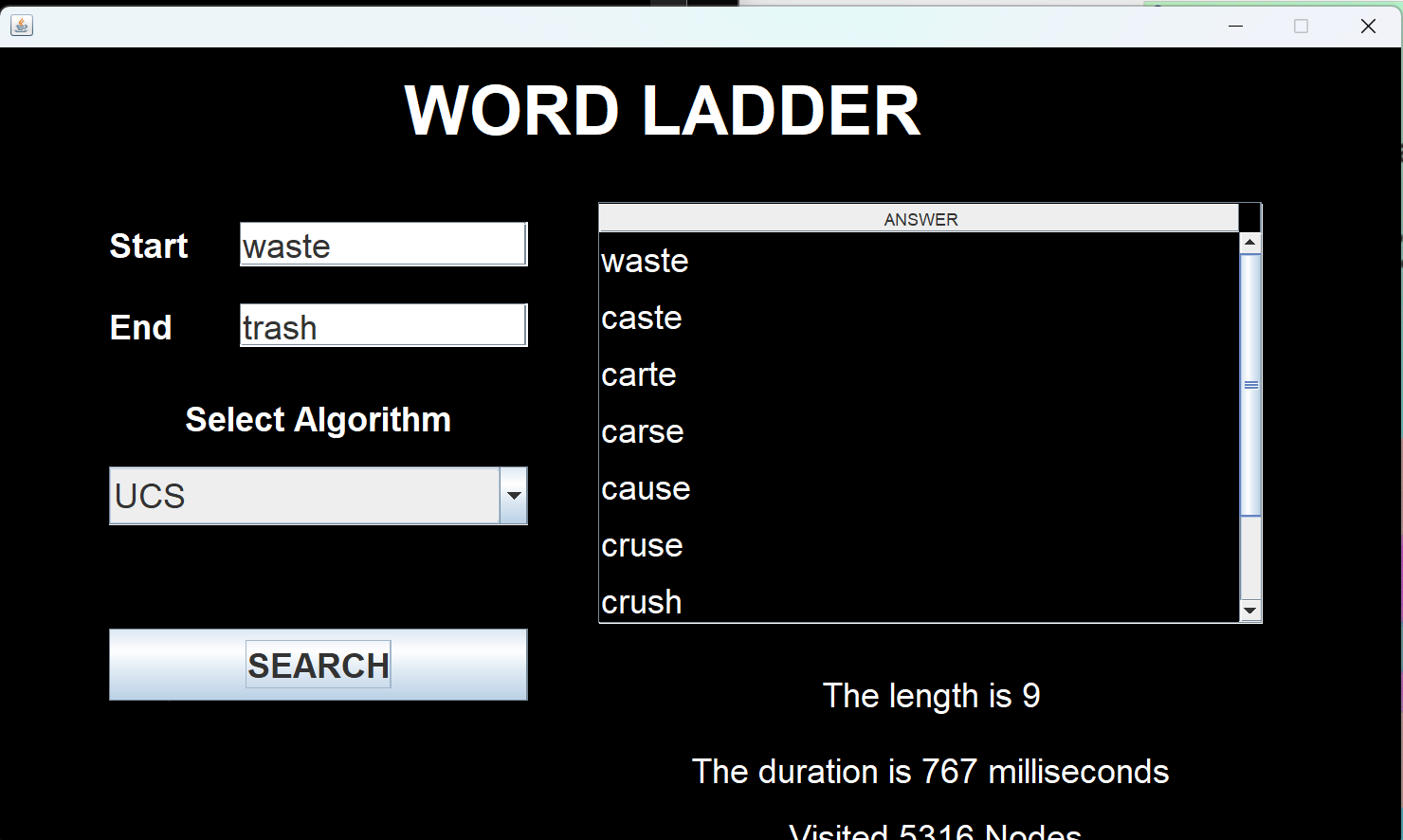
1. GBFS

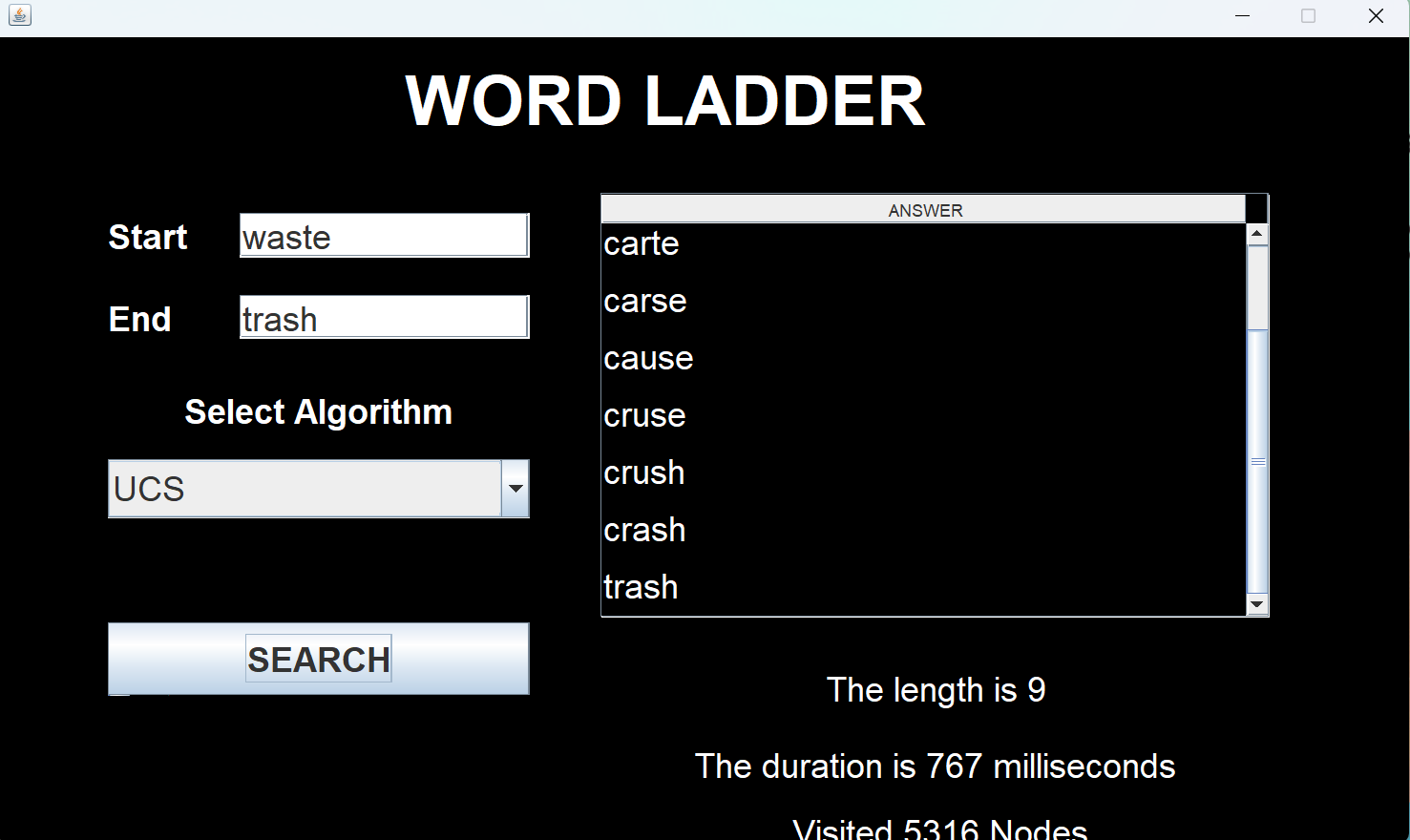


1. A Star

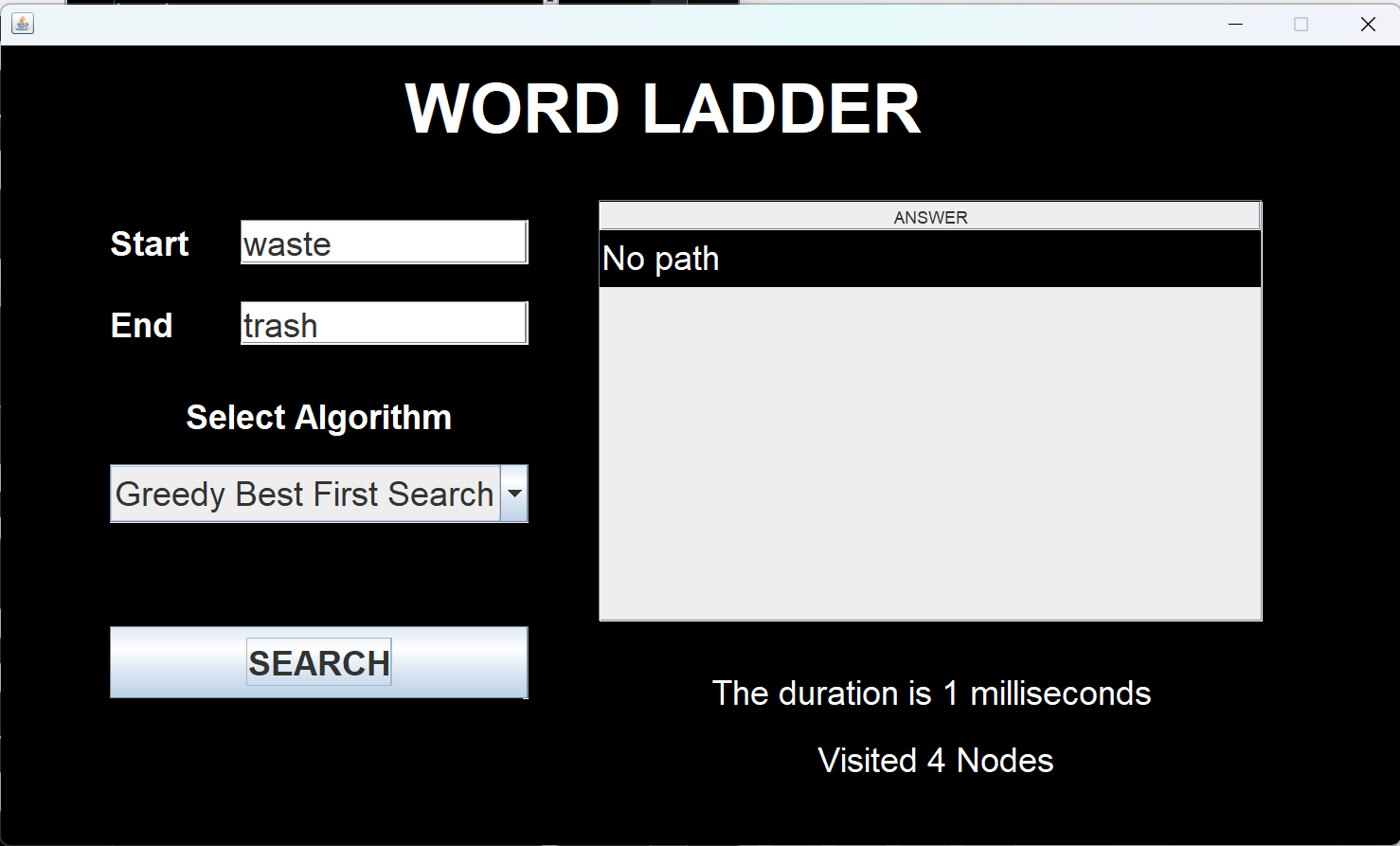


1. Waste – Trash
2. UCS

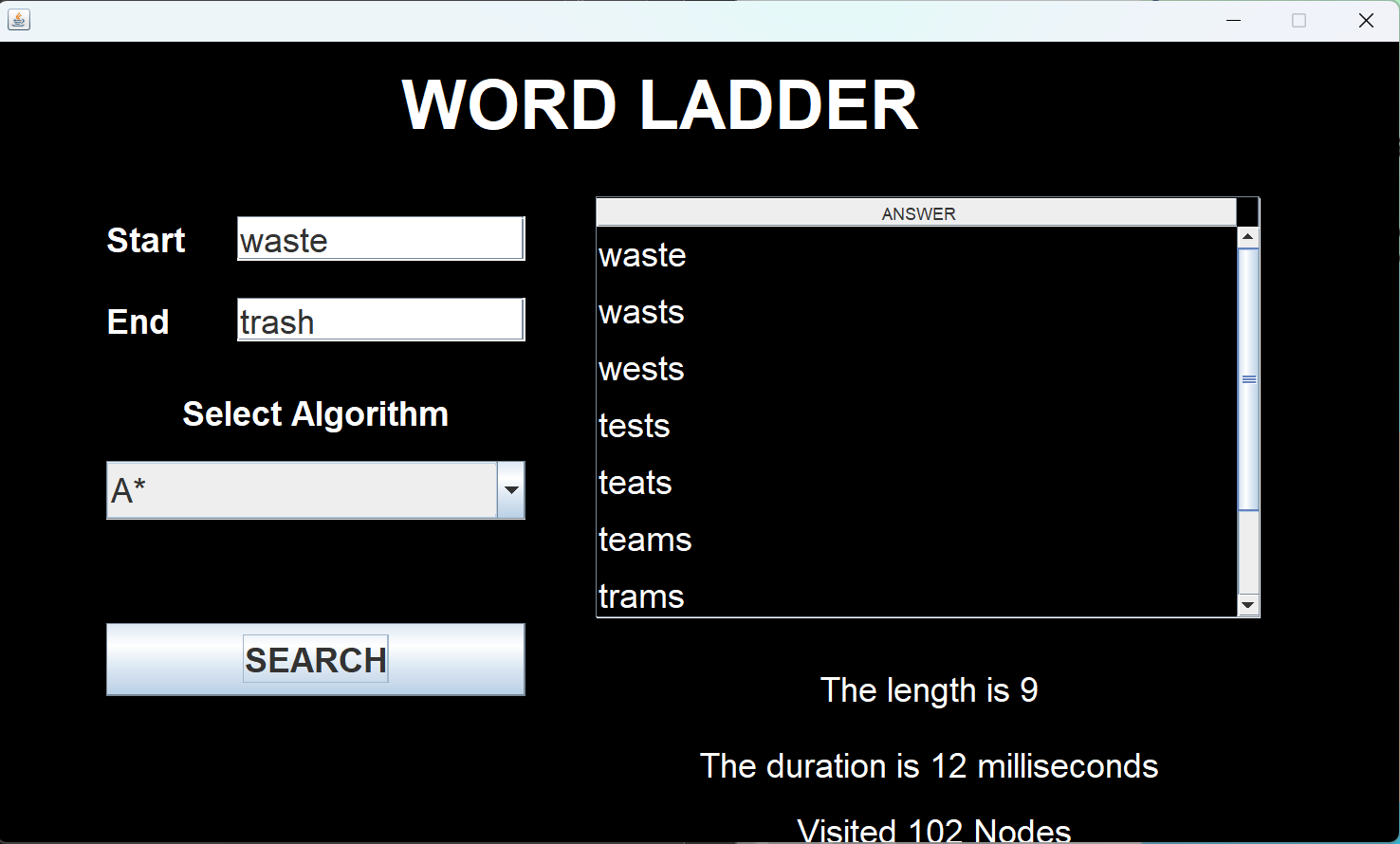


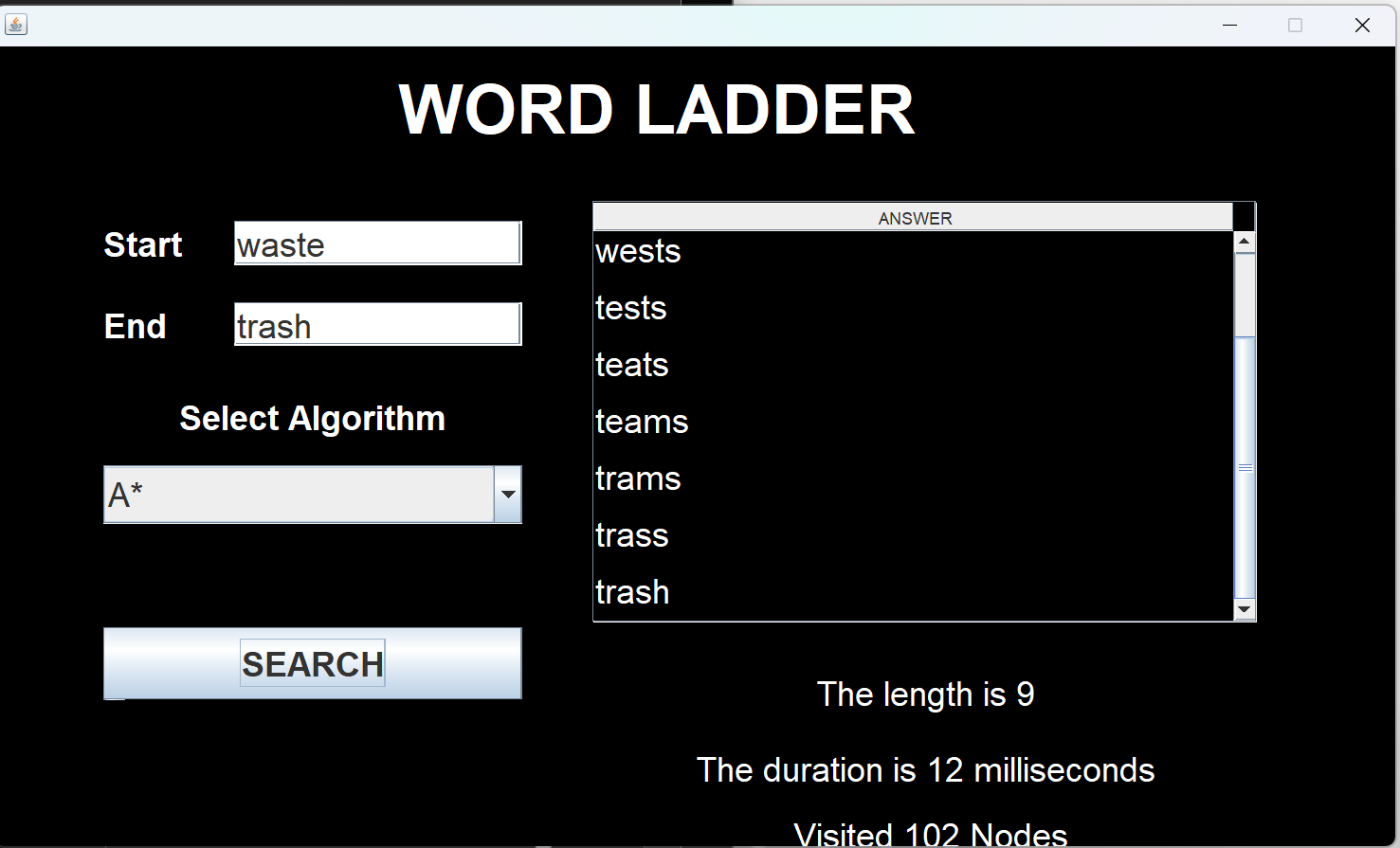


1. GBFS

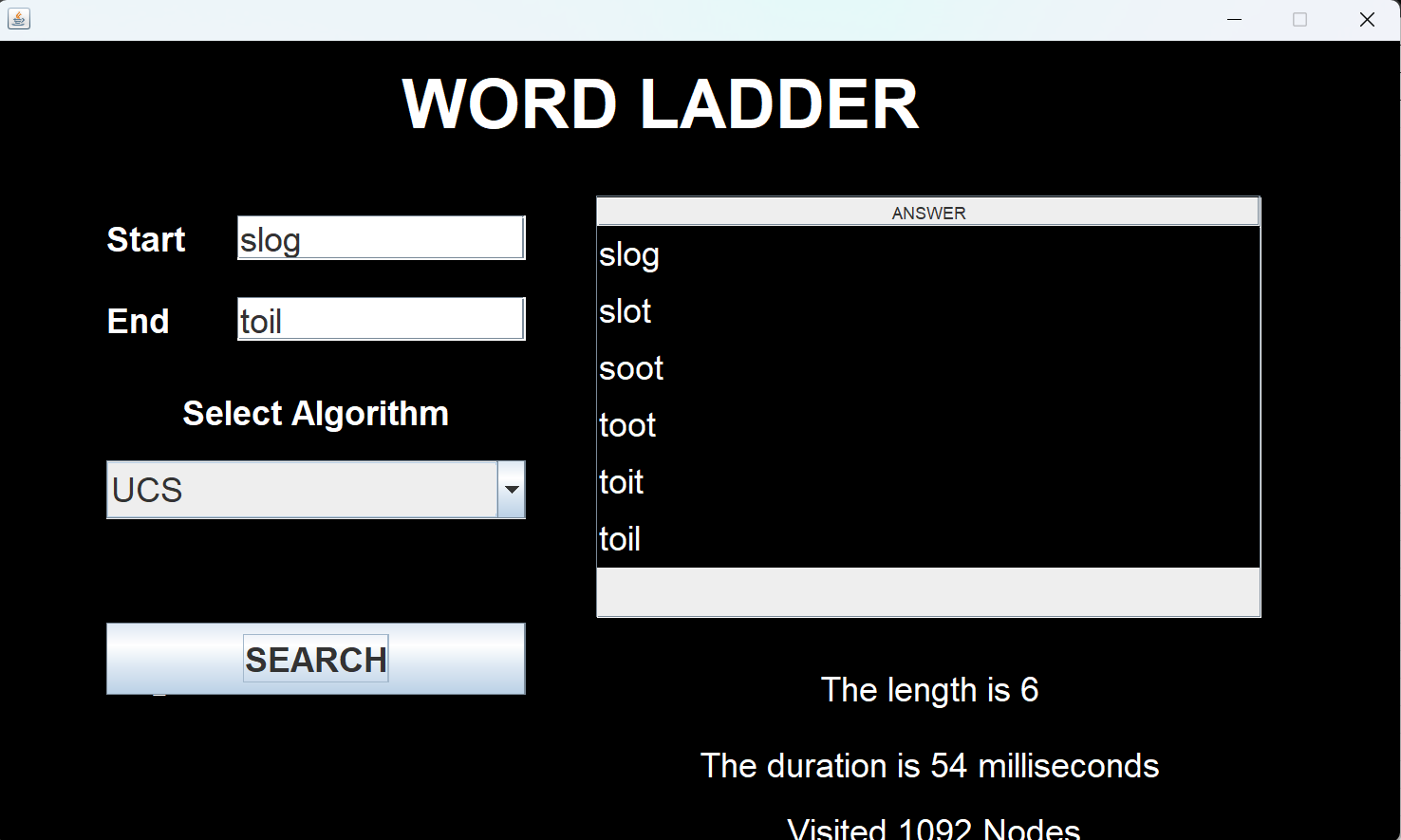


1. A Star

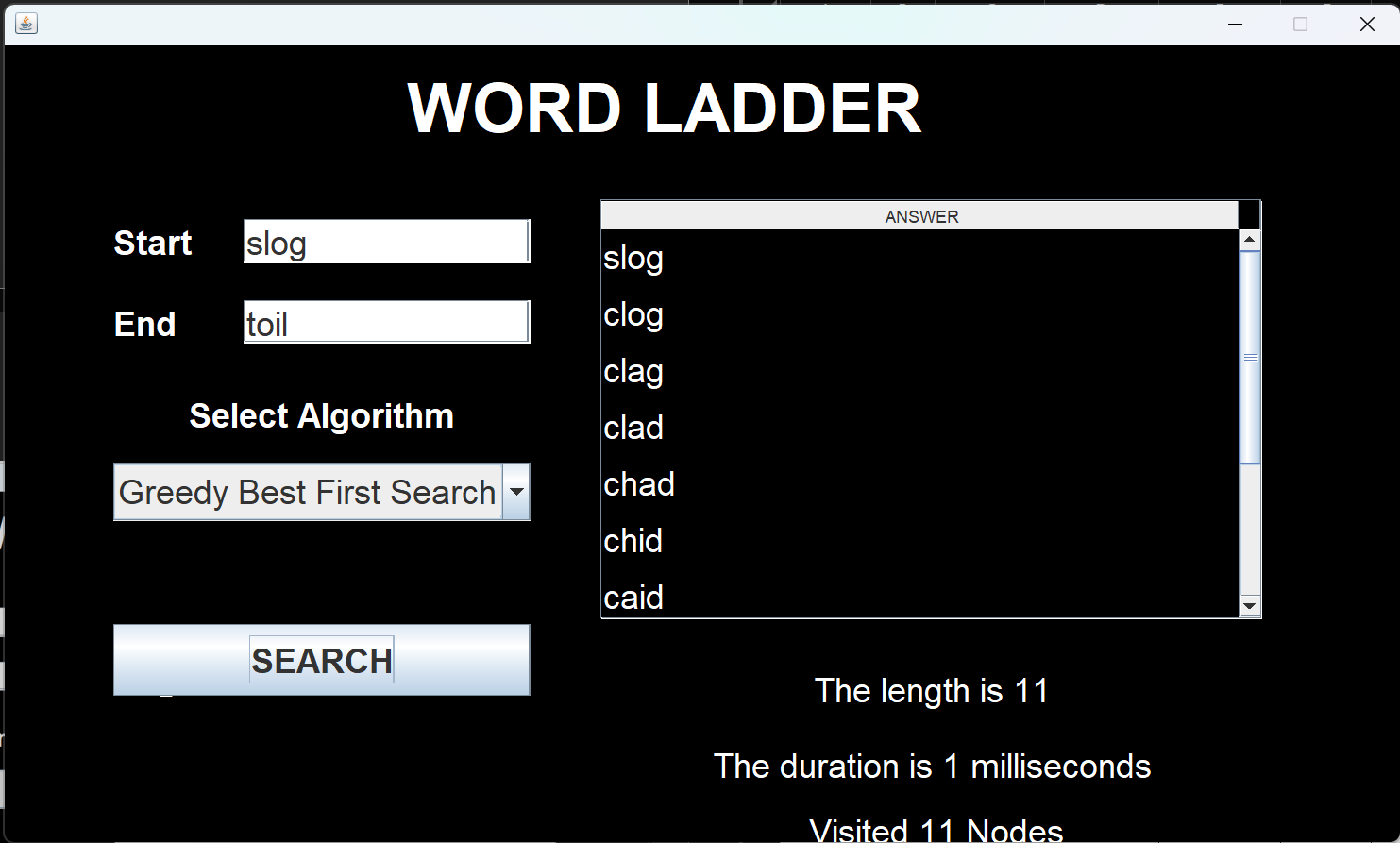




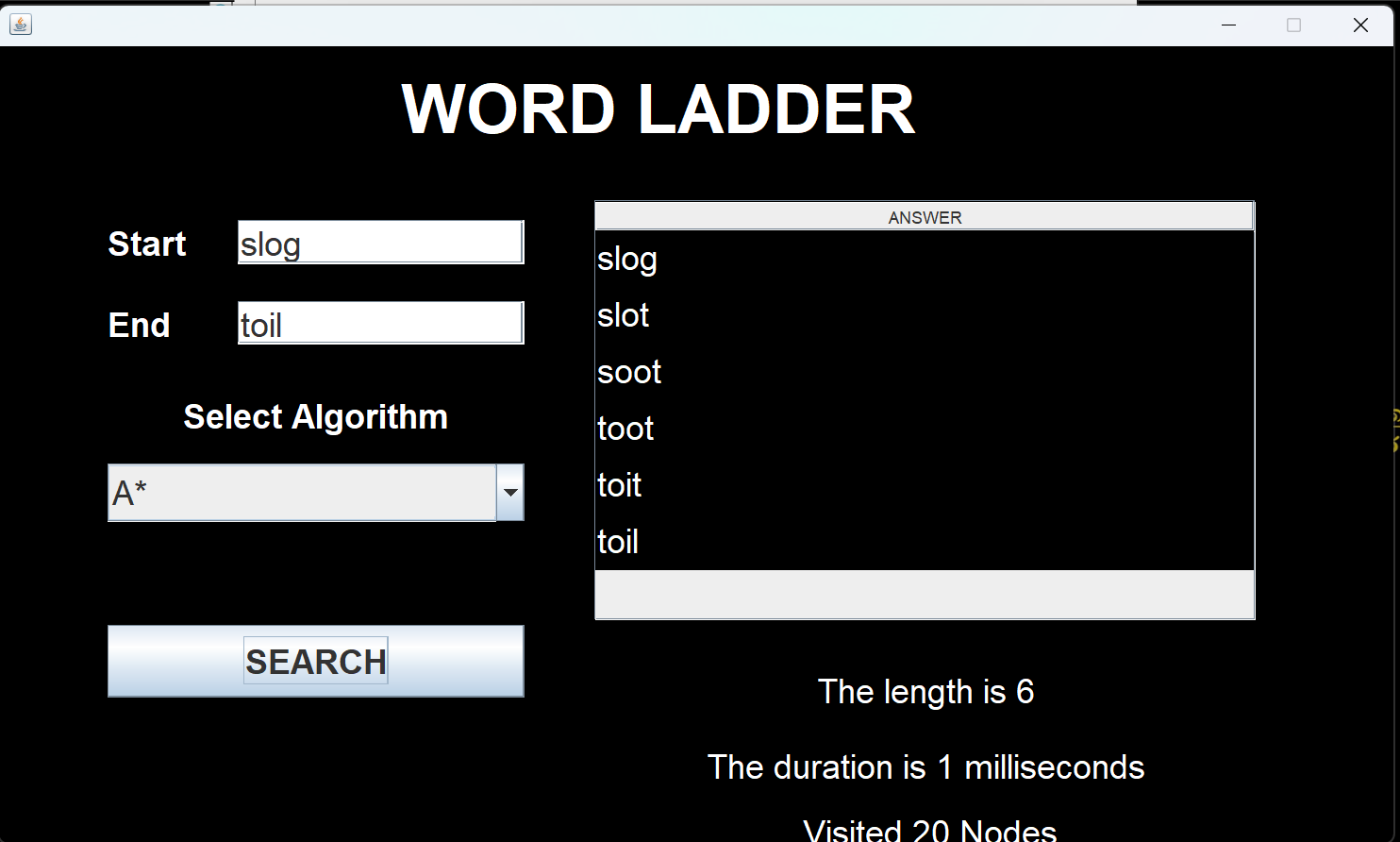
1. Slog – Toil
2. UCS



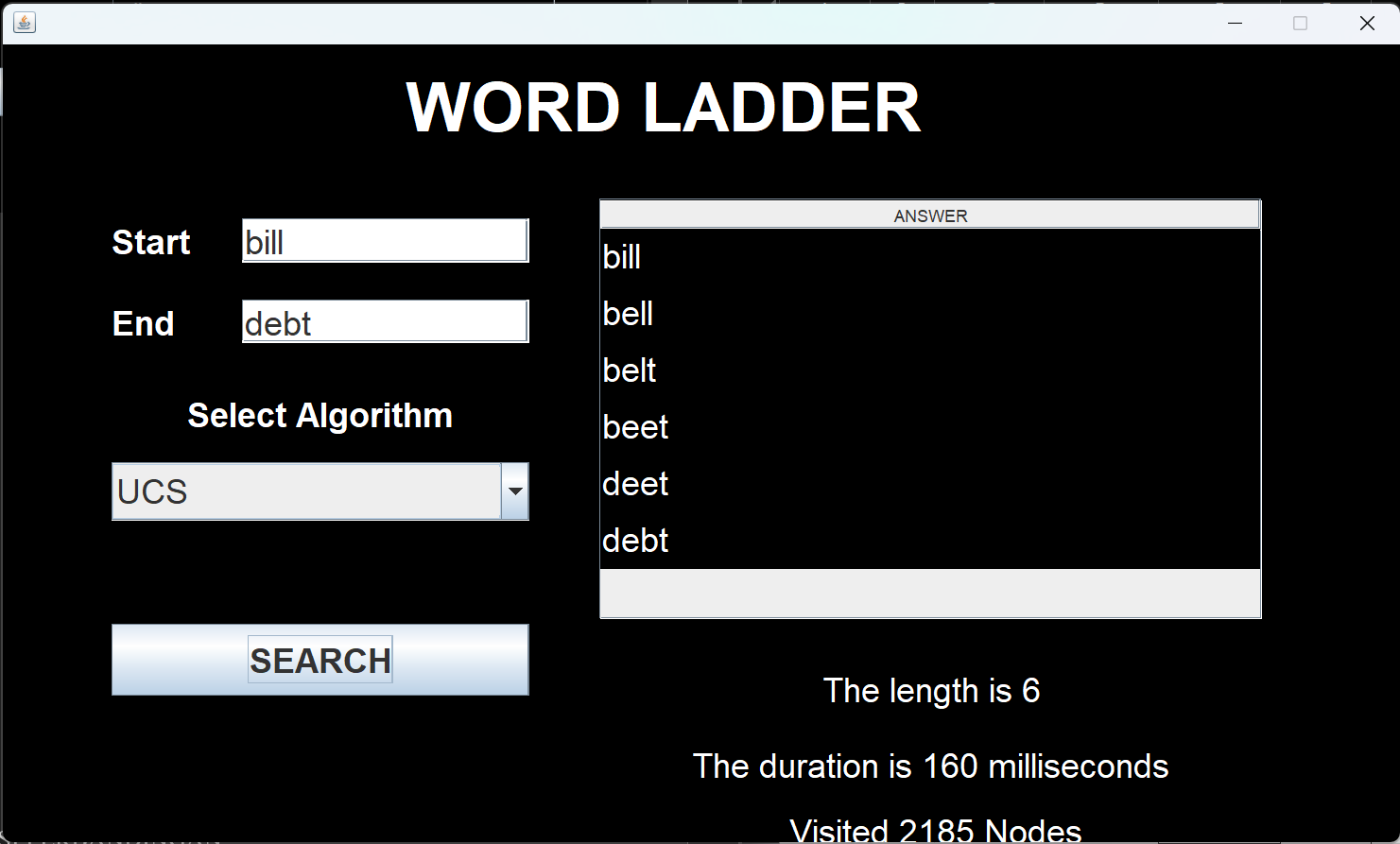
1. GBFS



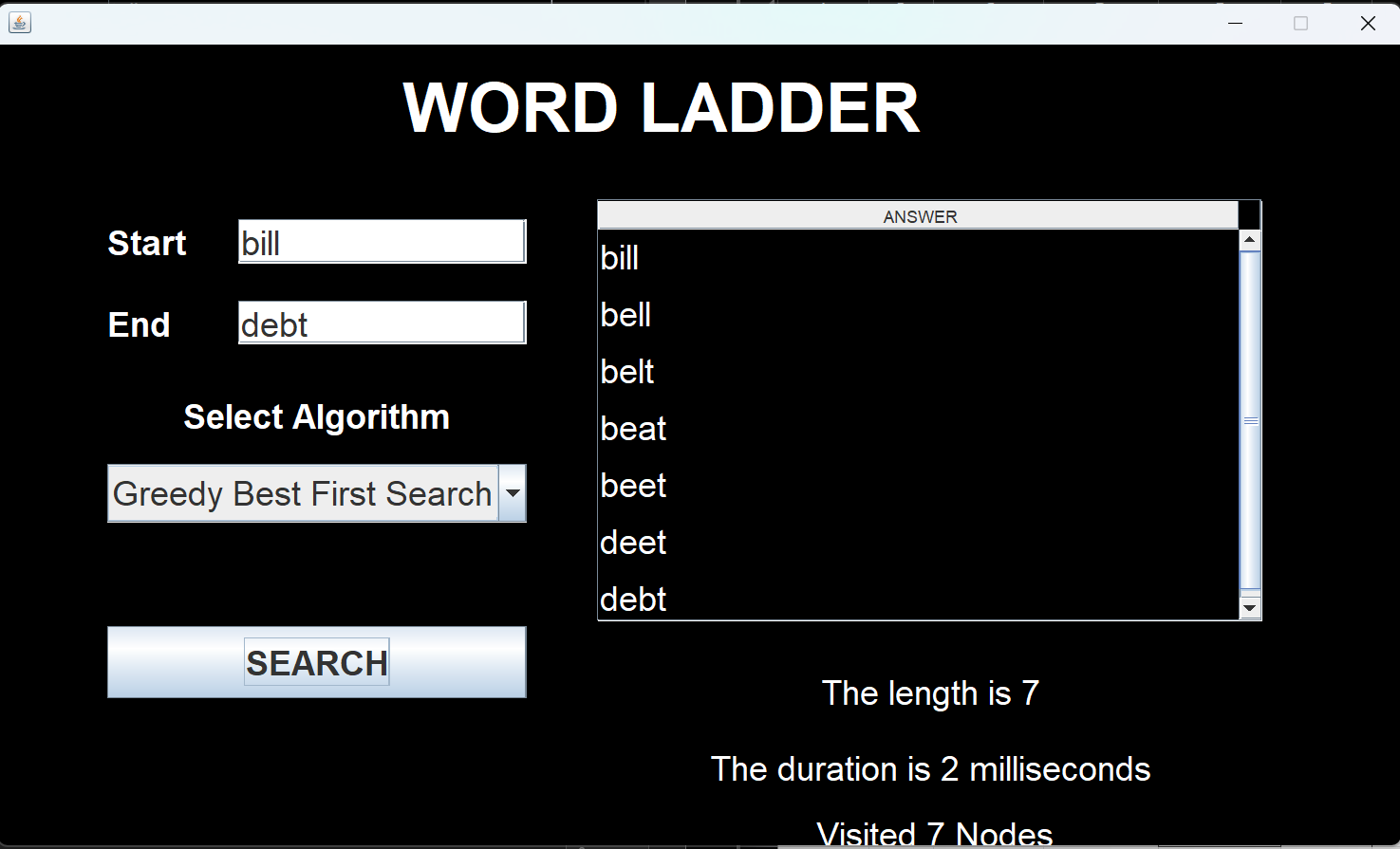
1. A Star



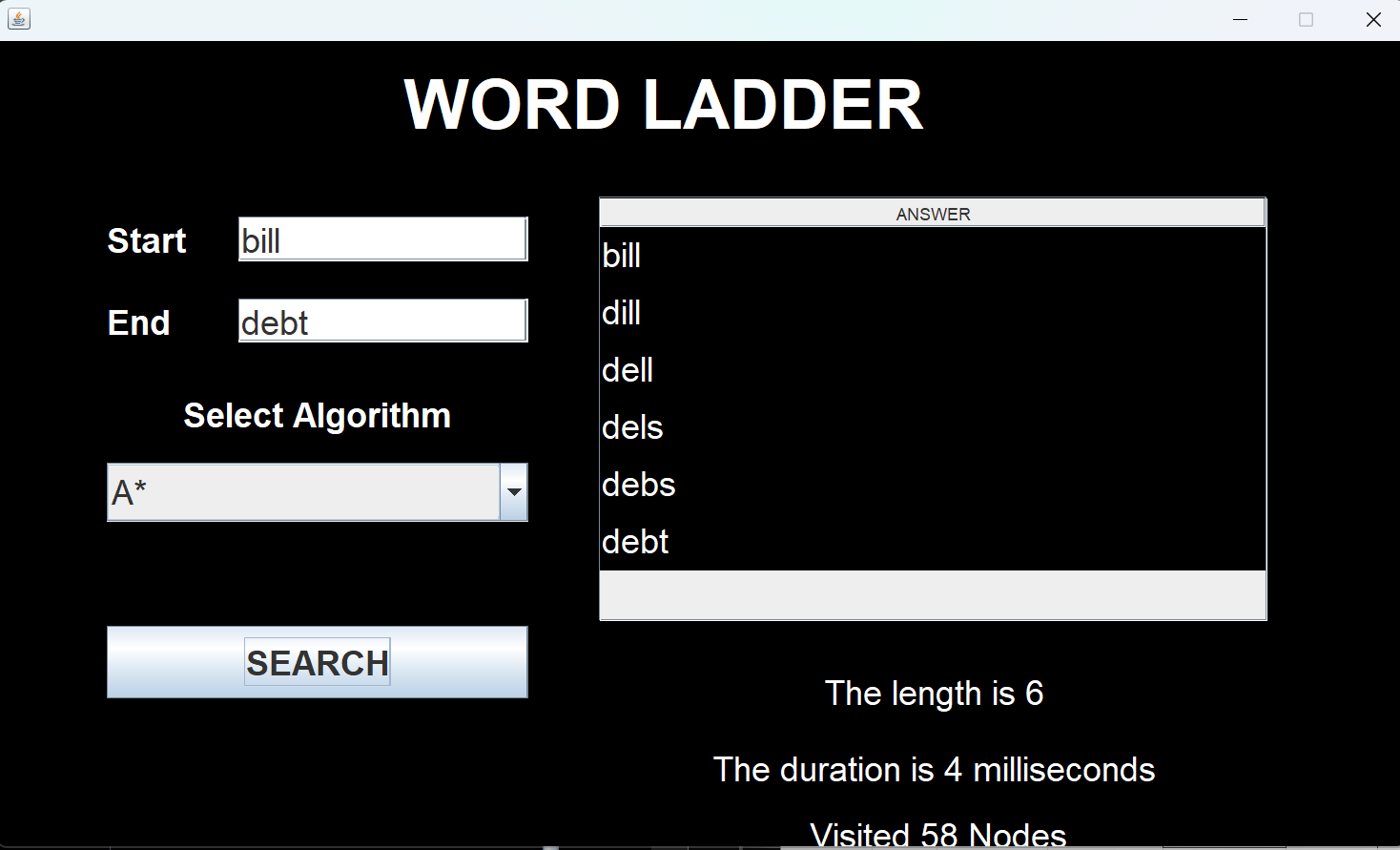
1. Bill – Debt
2. UCS



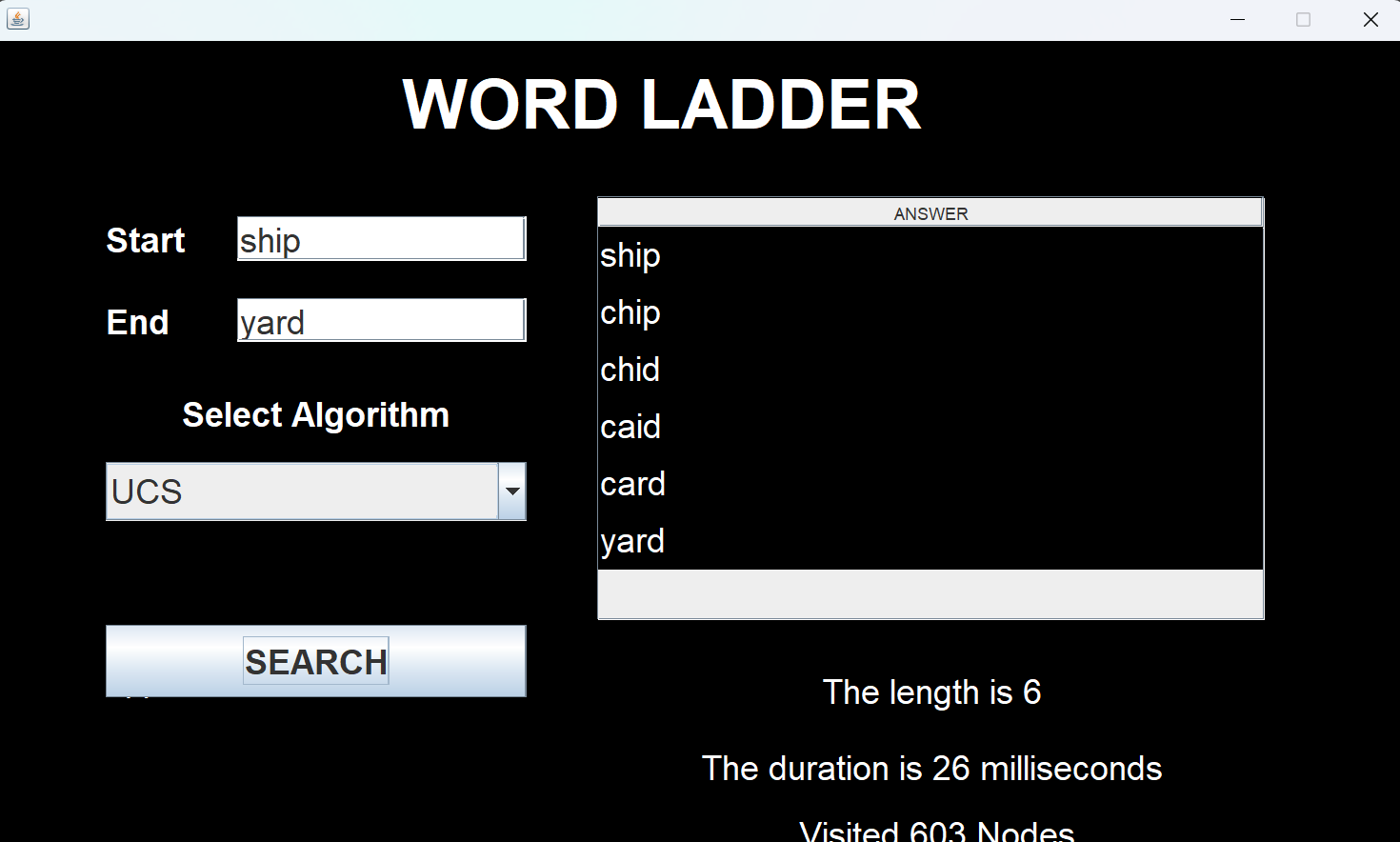
1. GBFS



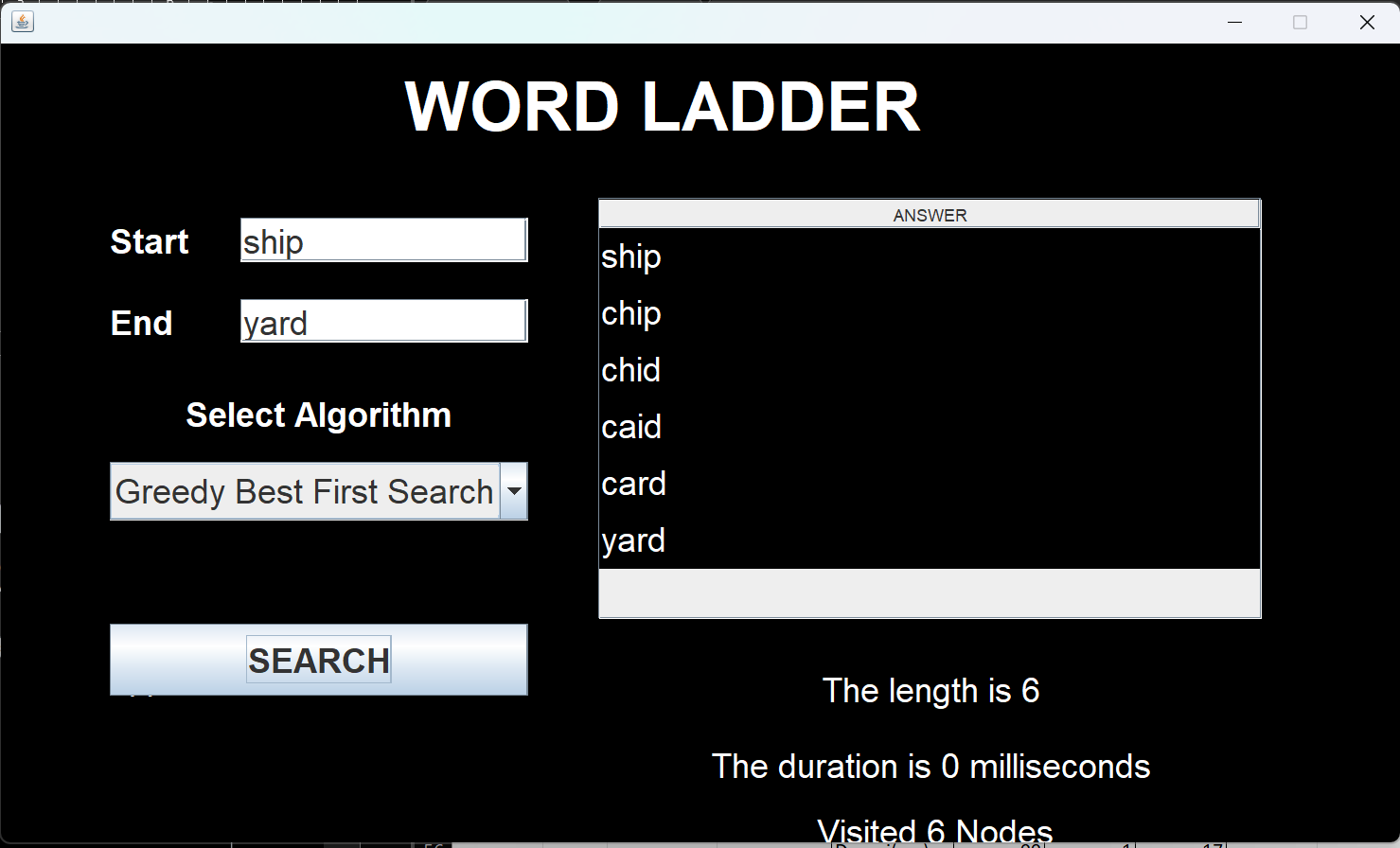
1. A Star



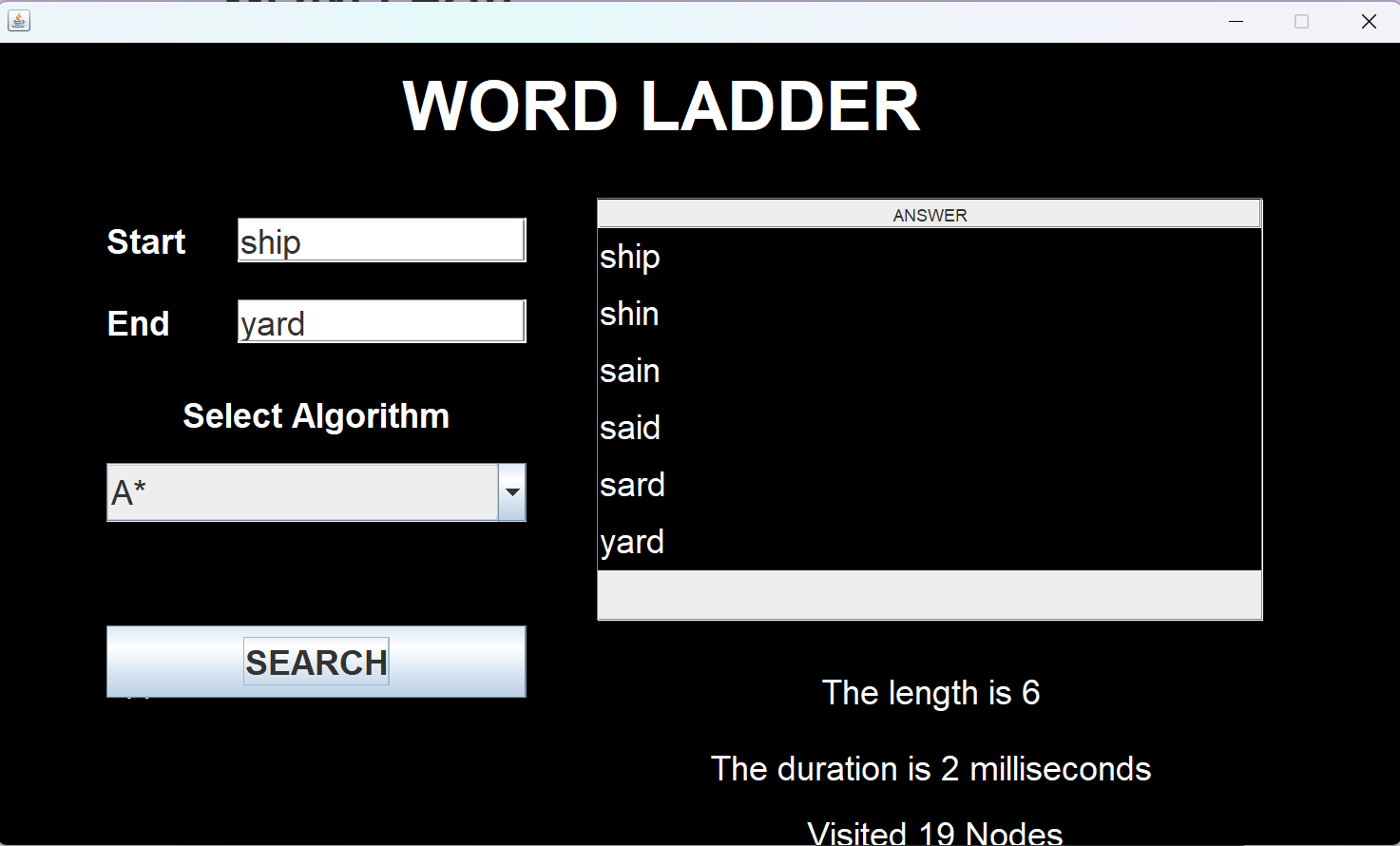
1. Ship – Yard
2. UCS



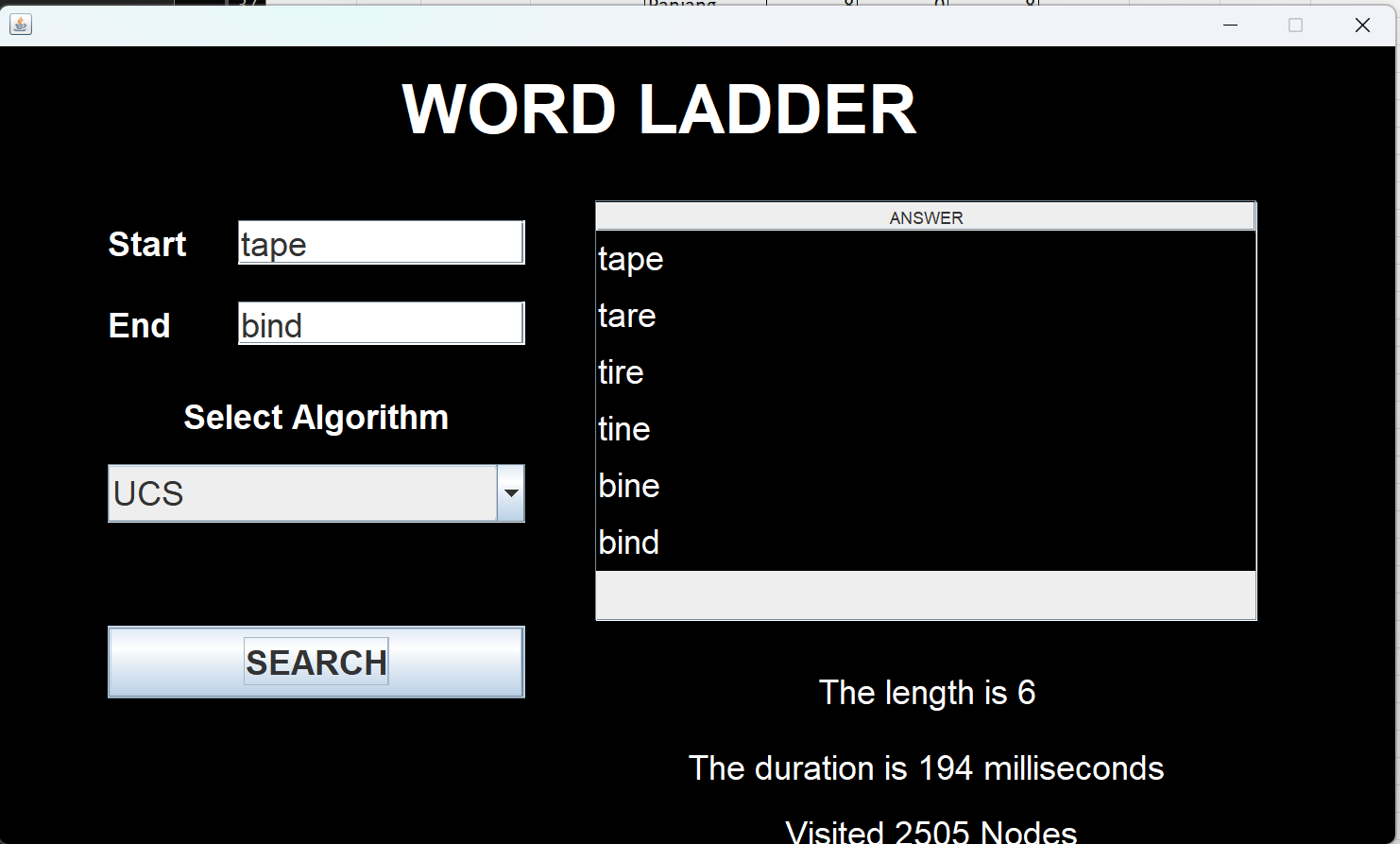
1. GBFS



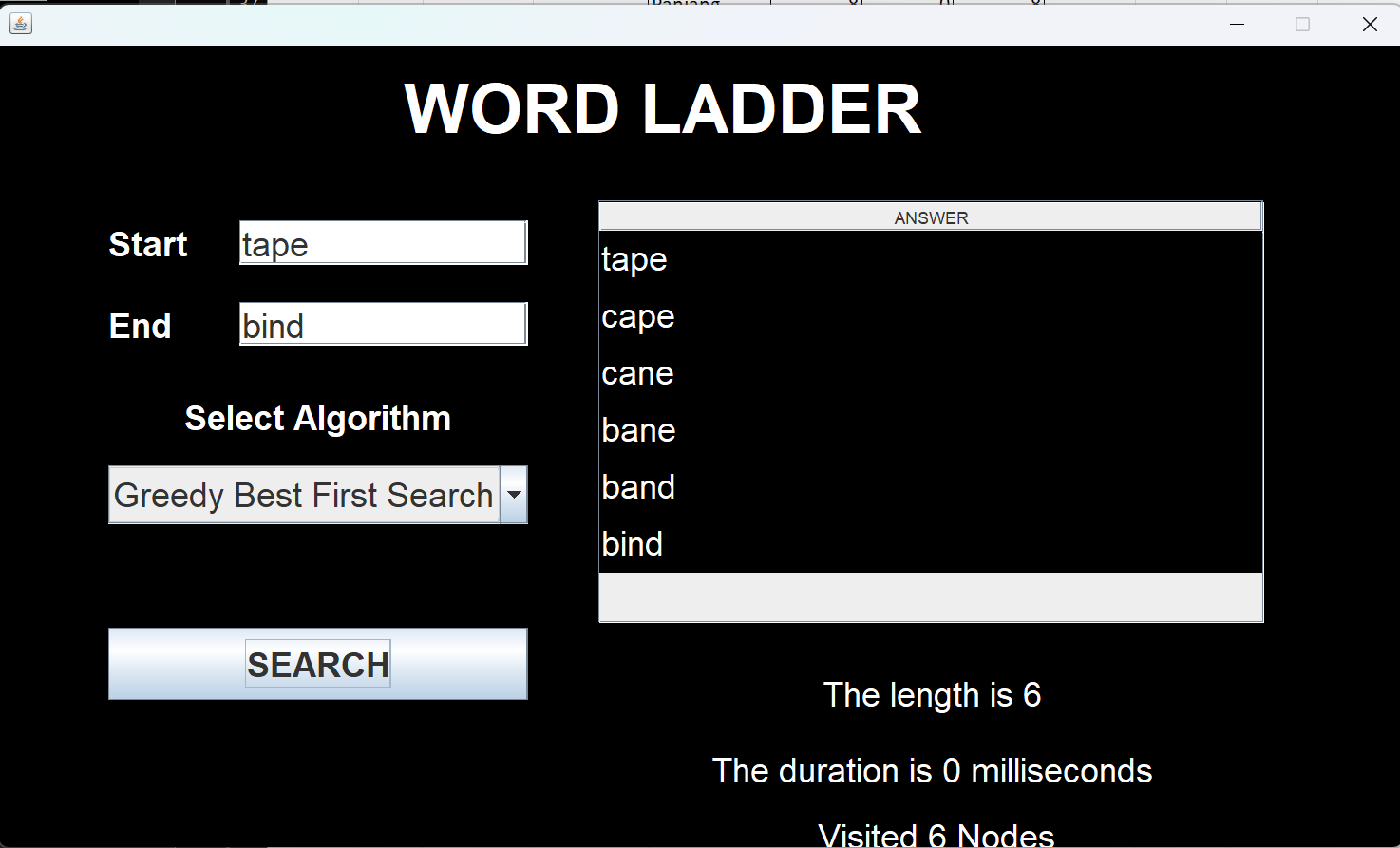
1. A Star



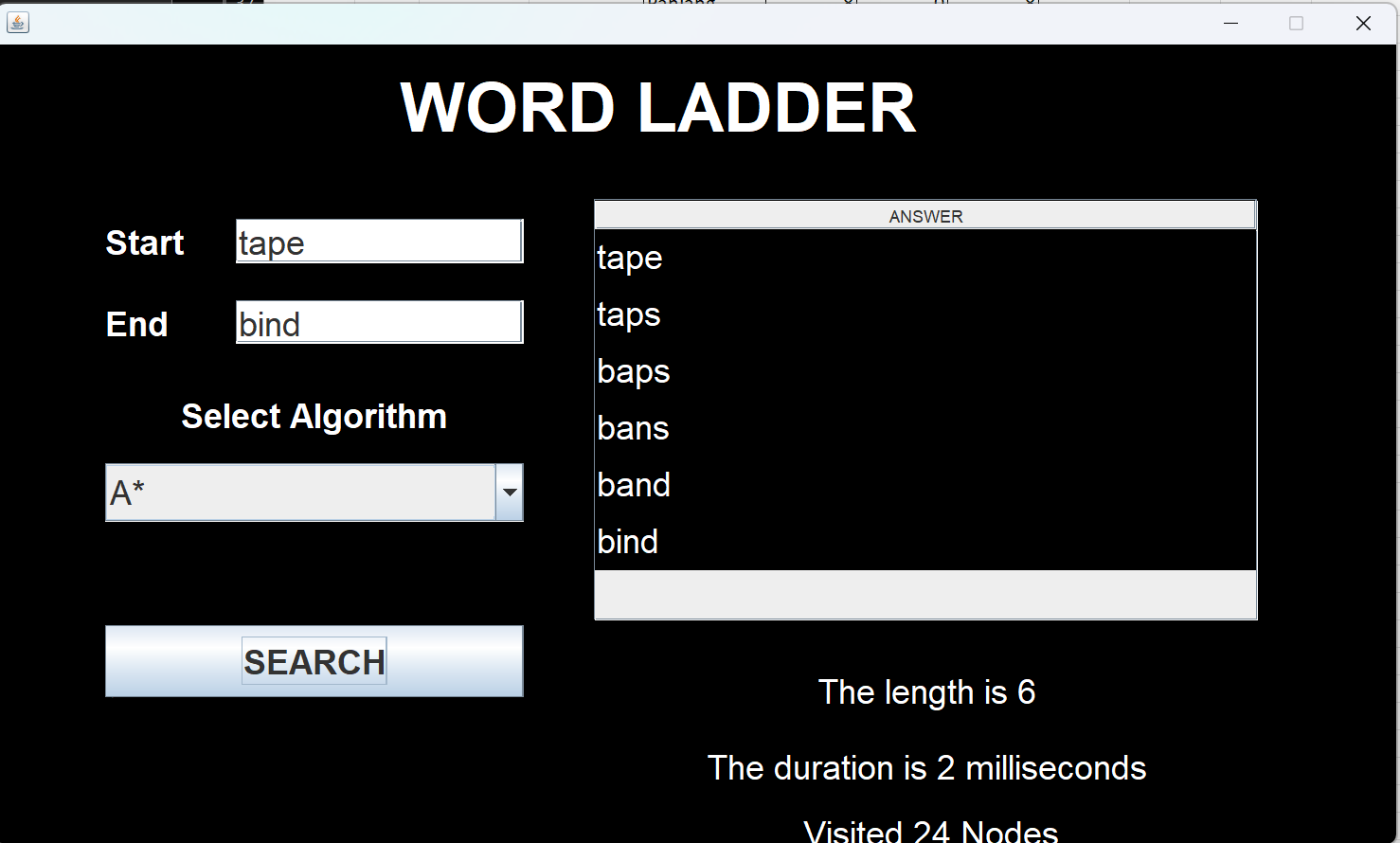
1. Tape – Bind
2. UCS



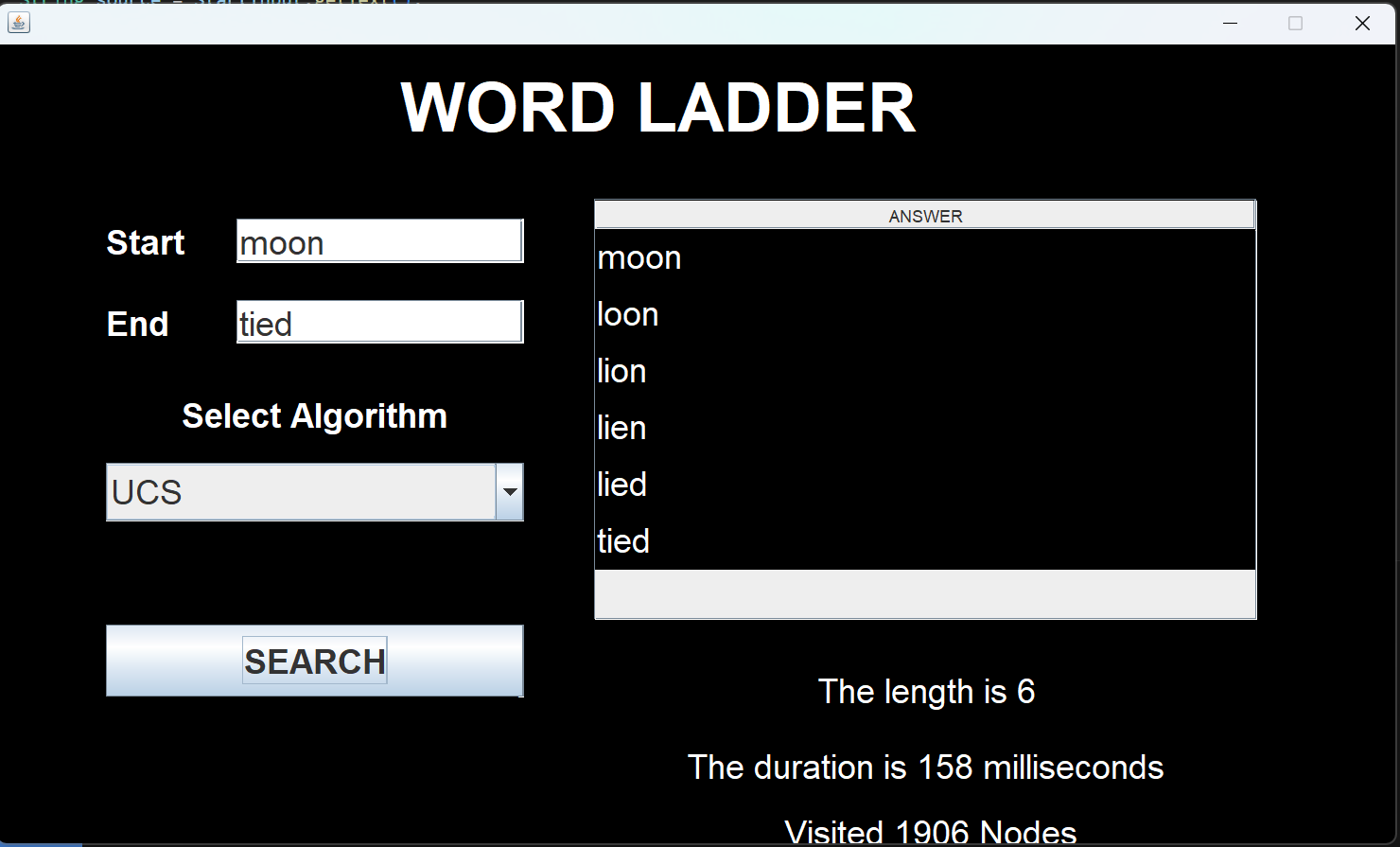
1. GBFS



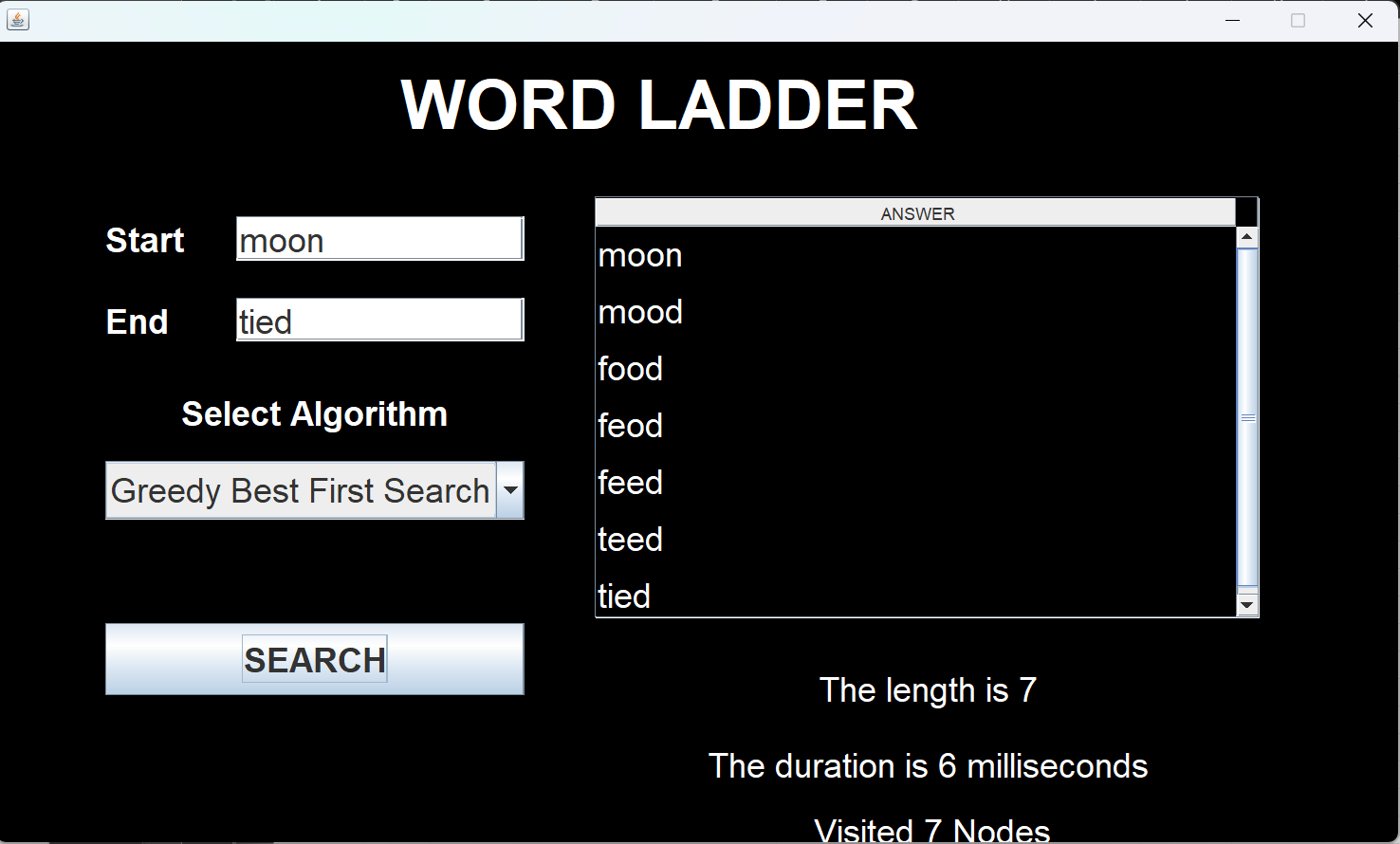
1. A Star



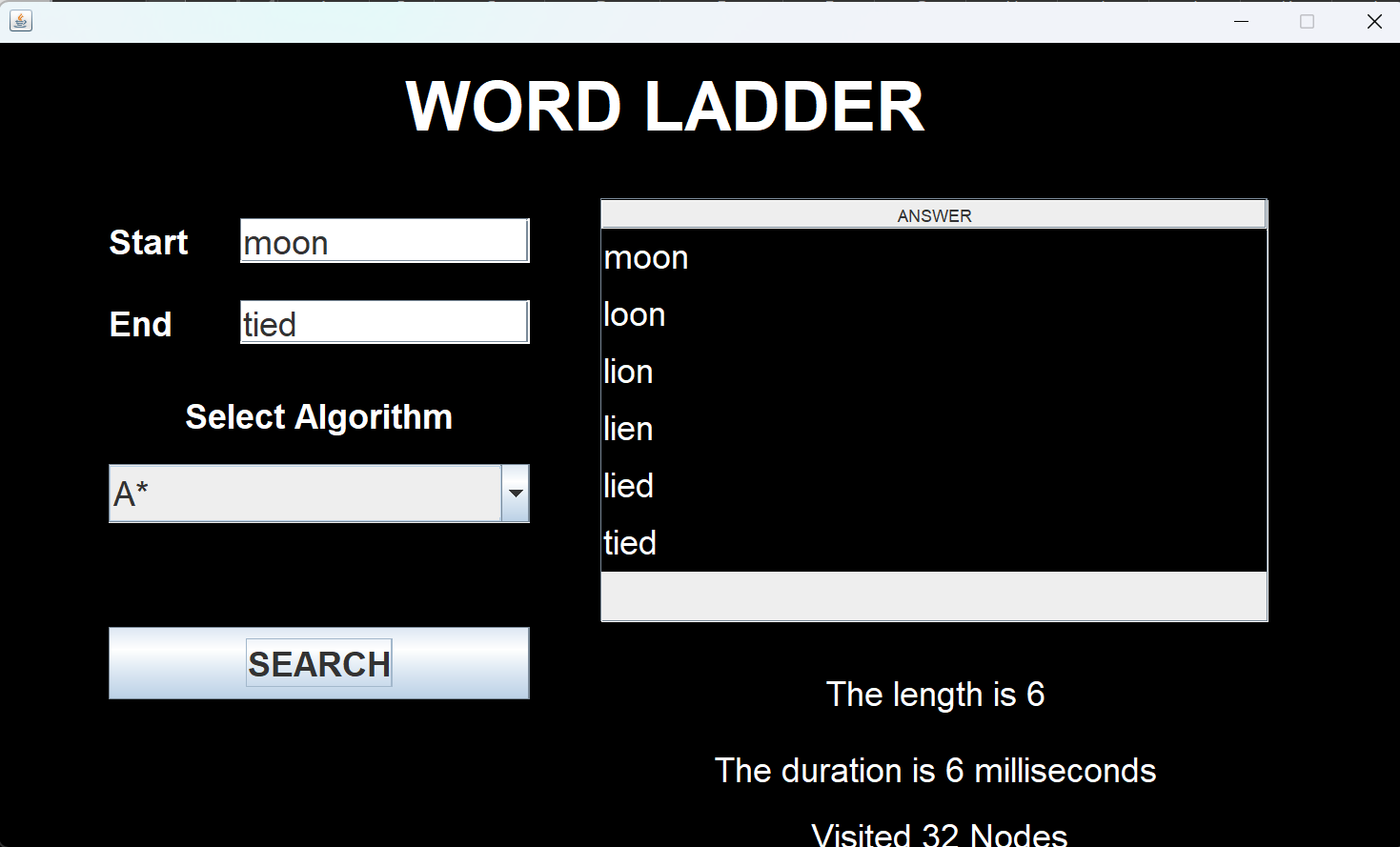
1. Moon – Tied
2. UCS



1. GBFS



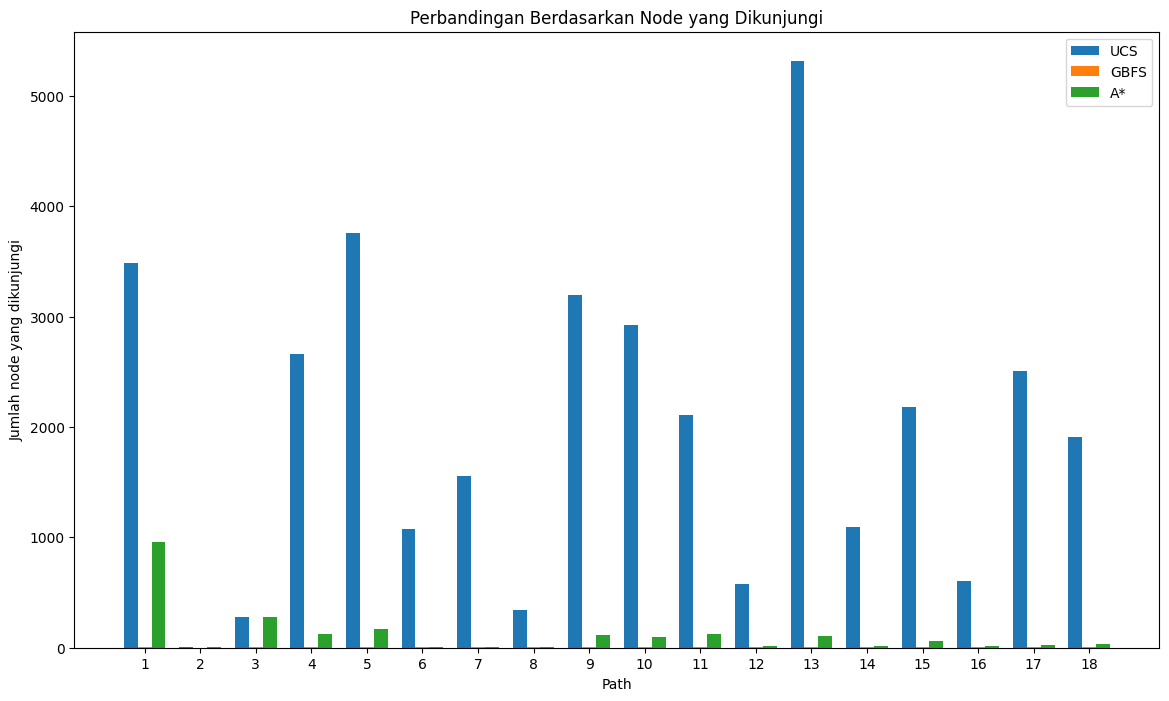
1. A Star

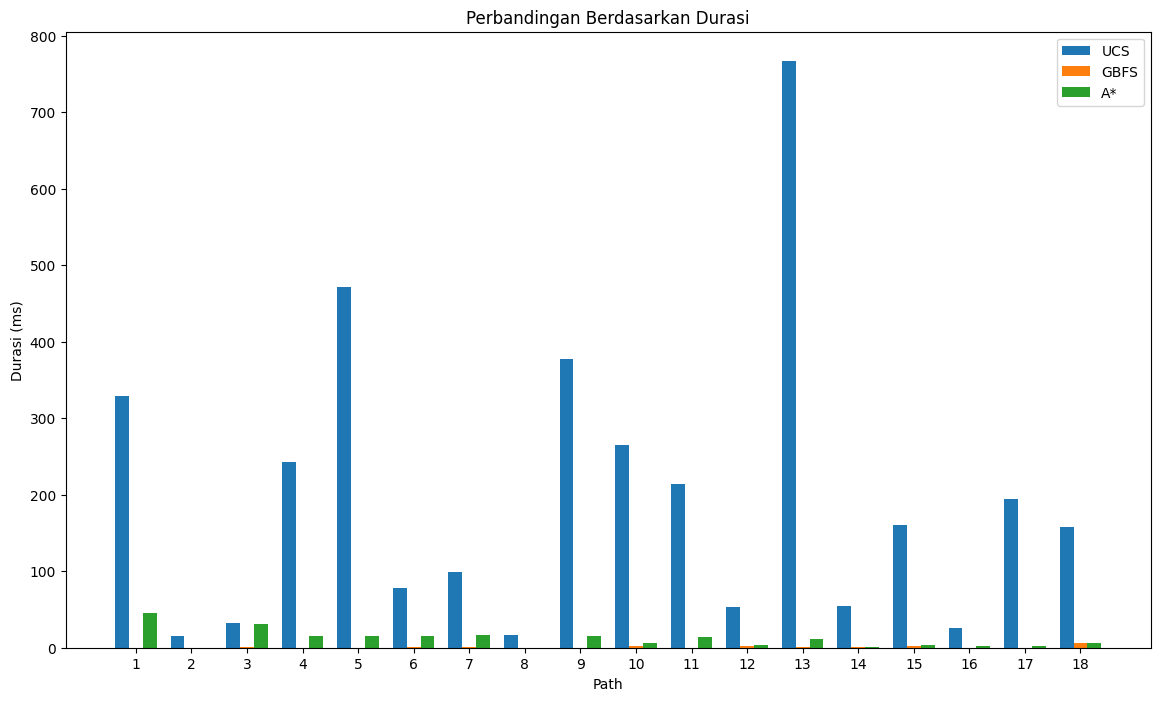


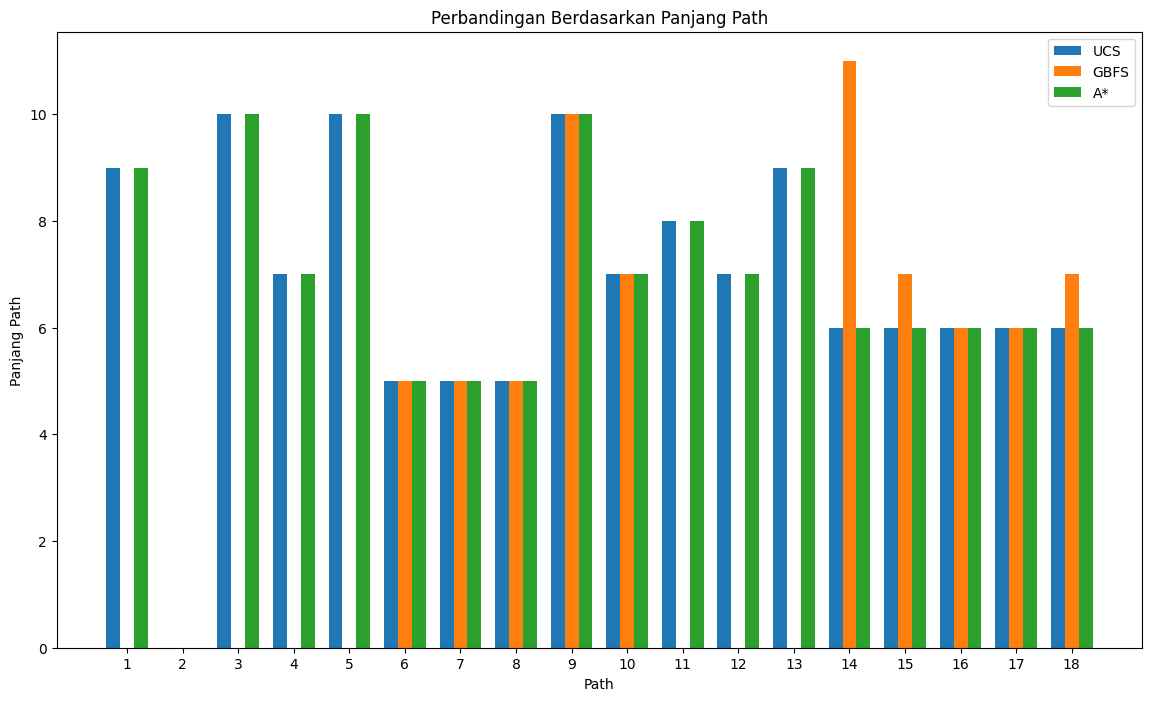
**BAGIAN IV**

**ANALISI PERBANDINGAN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Path** | | **Atribut** | **Algoritma** | | |
| **Start** | **End** | **UCS** | **GBFS** | **A\*** |
| **1** | baby | crib | **Panjang** | 9 | 0 | 9 |
| **Durasi(ms)** | 329 | 0 | 46 |
| **Kunjungan** | 3487 | 3 | 957 |
| **2** | unwarmed | upboiled | **Panjang** | 0 | 0 | 0 |
| **Durasi(ms)** | 15 | 0 | 0 |
| **Kunjungan** | 5 | 2 | 5 |
| **3** | frown | smile | **Panjang** | 10 | 0 | 10 |
| **Durasi(ms)** | 32 | 1 | 31 |
| **Kunjungan** | 278 | 7 | 278 |
| **4** | nice | mean | **Panjang** | 7 | 0 | 7 |
| **Durasi(ms)** | 243 | 0 | 15 |
| **Kunjungan** | 2658 | 4 | 126 |
| **5** | grain | wheat | **Panjang** | 10 | 0 | 10 |
| **Durasi(ms)** | 471 | 0 | 15 |
| **Kunjungan** | 3762 | 4 | 172 |
| **6** | feed | king | **Panjang** | 5 | 5 | 5 |
| **Durasi(ms)** | 78 | 1 | 15 |
| **Kunjungan** | 1075 | 5 | 5 |
| **7** | lame | duck | **Panjang** | 5 | 5 | 5 |
| **Durasi(ms)** | 99 | 1 | 17 |
| **Kunjungan** | 1559 | 5 | 8 |
| **8** | frog | pond | **Panjang** | 5 | 5 | 5 |
| **Durasi(ms)** | 17 | 0 | 0 |
| **Kunjungan** | 344 | 5 | 5 |
| **9** | crazy | party | **Panjang** | 10 | 10 | 10 |
| **Durasi(ms)** | 377 | 0 | 15 |
| **Kunjungan** | 3192 | 10 | 114 |
| **10** | slam | bang | **Panjang** | 7 | 7 | 7 |
| **Durasi(ms)** | 265 | 2 | 6 |
| **Kunjungan** | 2927 | 7 | 94 |
| **11** | trick | treat | **Panjang** | 8 | 0 | 8 |
| **Durasi(ms)** | 214 | 0 | 14 |
| **Kunjungan** | 2111 | 4 | 123 |
| **12** | knack | skill | **Panjang** | 7 | 0 | 7 |
| **Durasi(ms)** | 53 | 2 | 3 |
| **Kunjungan** | 581 | 6 | 20 |
| **13** | waste | trash | **Panjang** | 9 | 0 | 9 |
| **Durasi(ms)** | 767 | 1 | 12 |
| **Kunjungan** | 5316 | 4 | 102 |
| **14** | slog | toil | **Panjang** | 6 | 11 | 6 |
| **Durasi(ms)** | 54 | 1 | 1 |
| **Kunjungan** | 1092 | 11 | 20 |
| **15** | bill | debt | **Panjang** | 6 | 7 | 6 |
| **Durasi(ms)** | 160 | 2 | 4 |
| **Kunjungan** | 2185 | 7 | 58 |
| **16** | ship | yard | **Panjang** | 6 | 6 | 6 |
| **Durasi(ms)** | 26 | 0 | 2 |
| **Kunjungan** | 603 | 6 | 19 |
| **17** | tape | bind | **Panjang** | 6 | 6 | 6 |
| **Durasi(ms)** | 194 | 0 | 2 |
| **Kunjungan** | 2505 | 6 | 24 |
| **18** | moon | tied | **Panjang** | 6 | 7 | 6 |
| **Durasi(ms)** | 158 | 6 | 6 |
| **Kunjungan** | 1906 | 7 | 32 |







**BAGIAN V**

**IMPLEMENTASI GUI(*GRAPHICAL USER INTERFACE*)**

Padaa project ini, GUI yang dibuuat menggunakan kakas Java bernama Java Swing. Aplikasi Word Ladder adalah sebuah alat yang memungkinkan pengguna untuk menjelajahi serangkaian kata yang berbeda satu per satu, dengan setiap kata dalam serangkaian hanya berbeda satu huruf dari kata sebelumnya. Aplikasi ini dilengkapi dengan antarmuka pengguna yang ramah, memudahkan pengguna untuk melakukan pencarian Word Ladder dengan cepat dan efisien. Saat pertama kali aplikasi dibuka, pengguna disambut dengan antarmuka yang bersih dan terorganisir dengan baik. Judul besar "WORD LADDER" dengan tata letak yang menarik menjadi fokus utama, memberikan kesan yang jelas tentang tujuan aplikasi. Pengguna kemudian diminta untuk memasukkan kata awal dan akhir ke dalam kolom yang sesuai. Ada juga pilihan untuk memilih algoritma pencarian yang diinginkan, seperti Uniform Cost Search (UCS), Greedy Best First Search, atau A\*. Ini memberikan pengguna fleksibilitas dalam memilih pendekatan pencarian yang paling sesuai dengan kebutuhan mereka. Setelah memasukkan kata awal, kata akhir, dan memilih algoritma pencarian, pengguna cukup mengeklik tombol "SEARCH" untuk memulai proses pencarian. Selama proses ini, pengguna diberi umpan balik visual berupa label "ProcessingLabel", yang memberi tahu mereka bahwa operasi sedang berlangsung. Setelah pencarian selesai, hasilnya ditampilkan dalam bentuk tabel di bagian bawah antarmuka. Jika jalur dari kata awal ke kata akhir ditemukan, kata-kata dalam jalur akan ditampilkan dalam tabel dengan urutan yang sesuai. Namun, jika tidak ada jalur yang ditemukan, tabel akan menampilkan pesan yang sesuai. Selain itu, pengguna juga diberi informasi tambahan tentang pencarian, seperti durasi pencarian, jumlah node yang dikunjungi, dan panjang jalur. Hal ini membantu pengguna untuk memahami kinerja pencarian dan memberikan gambaran yang lebih jelas tentang hasilnya. Dalam semua kasus di atas, pesan kesalahan yang jelas dan informatif akan muncul di antarmuka pengguna untuk memberi tahu pengguna tentang masalah yang terjadi. Misalnya, ketika pengguna tidak memasukkan kata awal atau akhir, atau keduanya, dan mencoba untuk memulai pencarian, pesan kesalahan akan muncul meminta mereka untuk memasukkan kedua kata tersebut sebelum melanjutkan. Selain itu, jika kata awal atau akhir yang dimasukkan tidak ada dalam daftar kata yang tersedia, pengguna akan diberi tahu bahwa salah satu atau kedua kata tersebut tidak valid dan harus dipertimbangkan kembali. Tidak hanya itu, jika panjang kata awal dan akhir tidak sama, itu tidak akan mungkin membentuk Word Ladder antara dua kata tersebut. Dalam hal ini, pesan kesalahan akan ditampilkan untuk mengingatkan pengguna agar memasukkan kata dengan panjang yang sama. Terakhir, jika pengguna memasukkan karakter selain huruf alfabet (misalnya, angka, tanda baca, atau karakter khusus), pesan kesalahan akan muncul meminta mereka untuk memasukkan kata yang valid. Dengan antarmuka pengguna yang intuitif, proses pencarian yang cepat, dan umpan balik yang informatif, aplikasi Word Ladder ini menjadi alat yang berguna bagi siapa saja yang tertarik untuk menjelajahi dunia kata-kata dengan cara yang menarik dan interaktif.

**BAGIAN VI**

**REPOSITORY GITHUB**

<https://github.com/FrancescoMichael/Tucil1_13522038.git>

BAGIAN V

CHECKLIST

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dijalankan. | ✓ |  |
| 1. Program dapat menemukan rangkaian kata dari start word ke end word sesuai aturan permainan dengan algoritma UCS | ✓ |  |
| 1. Solusi yang diberikan pada algoritma UCS optimal | ✓ |  |
| 1. Program dapat menemukan rangkaian kata dari start word ke end word sesuai aturan permainan dengan algoritma Greedy Best First Search | ✓ |  |
| 1. Program dapat menemukan rangkaian kata dari start word ke end word sesuai aturan permainan dengan algoritma A\* | ✓ |  |
| 1. Solusi yang diberikan pada algoritma A\* optimal | ✓ |  |
| 1. [Bonus]: Program memiliki tampilan GUI | ✓ |  |