アルゴリズム論2 レポート

276156 根本 貴大

1. 各種アルゴリズムのプログラム作成

線形探索

#include <stdio.h>

int lin\_search(int a[], int n, int key);

int lin\_search\_s(int x[], int n, int key);

#define NUM 1000

#pragma warning(disable:4996)

int count1, count2;

int main(void) {

int i, ky, idx;

int x[NUM + 1]; /\* 番兵法対策 \*/

char fname[100];

FILE \*fp;

printf("Input file name:");

scanf("%s", fname);

fopen\_s(&fp, fname, "r");

if (fp == NULL) {

printf("%sファイルが開けません\n", fname);

return -1;

}

for (i = 0; i < NUM; i++)

fscanf(fp, "%d", &x[i]);

printf("Number to search:");

scanf\_s("%d", &ky);

idx = lin\_search(x, NUM, ky);

if (idx == -1)

printf("Searching was failed\n");

else

printf("%d is located at %d \n", ky, idx);

printf("試行回数：%d \n", count1);

return 0;

}

int lin\_search(int a[], int n, int key) {

int i = 0;

while (1) {

if (i == n) {

count1++;

return(-1);

}

else

count1++;

if (a[i] == key) {

count2++;

return (i);

}

else

count2++;

i++;

}

}

/\*番兵\*/

int lin\_search\_s(int x[], int n, int key) {

int i = 0;

x[n] = key; /\* sentinel \*/

while (1) {

if (x[i] == key) {

count2++;

break;

}

else

count2++;

i++;

}

if (i == n) {

count1++;

return(-1);

}

else

count1++;

return i;

}

　2分探索

#include<stdio.h>

#include<stdlib.h>

#define NUM 1000

#pragma warning(disable:4996)

int bin\_search(int a[], int n, int key);

int bubble(int a[], int n);

int count = 0;

int main(void) {

int i, key, idx;

int x[NUM + 1]; /\* 番兵法対策 \*/

FILE \*fp;

char fname[64];

printf("Input file name:");

scanf("%s", fname);

fopen\_s(&fp, fname, "r");

if (fp == NULL) {

　printf("%sファイルを開けません\n", fname);

　return -1;

}

for (i = 0; i<NUM; i++)

fscanf(fp, "%d", &x[i]);

bubble(x, NUM); /\* bubblesort \*/

printf("Number to search:");

scanf("%d", &key);

idx = bin\_search(x, NUM, key);

if (idx == -1)

printf("Searching was failed!\n");

else

printf("%d is located %d \n", key, idx);

printf("試行回数%d\n\n", count);

return 0;

}

int bubble(int x[], int n) {

int i, j, tmp, count = 0;

for (i = 0; i < n - 1; i++) {

for (j = n - 1; j > i; j--) {

if (x[j - 1] > x[j]) {

tmp = x[j];

x[j] = x[j - 1];

x[j - 1] = tmp;

count++;

}

}

}

return count;

}

int bin\_search(int a[], int n, int key) {

int pl = 0;

int pr = n - 1;

int pc;

do {

pc = (pl + pr) / 2;

if (a[pc] == key)

return(pc);

else if (a[pc] < key) {

pl = pc + 1;

count++;

}

else {

pr = pc - 1;

count++;

}

} while (pl <= pr);

return -1;

1. 結果

線形探索

|  |  |  |  |
| --- | --- | --- | --- |
|  | 検索の場所 | count1 | count2 |
| test1.txt | 199 | 1 | 200 |
| test2.txt | 399 | 1 | 400 |
| test3.txt | 599 | 1 | 600 |
| test4.txt | 799 | 1 | 800 |

2分探索

|  |  |  |  |
| --- | --- | --- | --- |
|  | 検索の場所 | count1 | count2 |
| test1.txt | 500 | 10 | 0 |
| test2.txt | 500 | 10 | 0 |
| test3.txt | 500 | 10 | 0 |
| test4.txt | 500 | 10 | 0 |

1. 考察

線形探索のcountの平均値を求めると

　　(200 + 400 + 600 + 800) / 4 = 500となり、

線形探索のcountの平均値は500。

線形探索の時間計算量はO(nｍ)なので、線形探索の時間計算量はO(500) = 500となる。

二分探索のcountの平均値を求めると、二分探索の平均値は10。

二分探索の時間計算量はO(log2N)なので、O(log210) = 0.30103となる。