

Junior 2: Baywatch

Symbroson
Team-ID: 00165

9. September 2018

Inhaltsverzeichnis

1 Lösungsidee

Prinzipiell funktioniert diese Aufgabe wie eine Drehscheibe, bei der die eine Scheibe so gedreht werden muss, dass die gegebenen Symbole des Innenrings mit denen des Äußeren übereinstimmen. Dabei kann man alle nicht gegebenen Symbole vernachlässigen.

2 Umsetzung

Da die Ausgabe hier nicht in Worten erfolgen muss kann man sich das Parsen der Eingabedatei sparen. Ich lese also nur die ersten beiden Zeilen ein und speichere mir die Indizes aller numerischen Zeichen der Longstock-Liste die später verglichen werden sollen in ein Array. Wurden keine gefunden, wird der gesamte folgende Prozess ausgelassen und eine dementsprechende Information ausgegeben.

Falls doch, wird die George-Liste systematisch nach dem ersten numerischen Zeichen der Longstock-Liste abgesucht und ausgehend von dieser Position das weitere Muster validiert. Stimmen alle numerischen Zeichen der Longstock-Liste mit der George-Liste überein wird die George-Liste von dort aus ausgegeben. Dies wird so lange wiederholt, bis die nächste Suchposition die Länge der Liste(n) überschreitet, um alle möglichen Varianten zu erfassen.

3 Beispiele

3.1 baywatch1.txt

Hier würde es sich anbieten rückwärts zu suchen, da die Lösung genau bei -1 liegt (wenn man die Longstock-Liste dreht). Das macht aber im allgemeinen Fall keinen Unterschied, da das Ergebnis so oder so in der gleichen Zeitkomplexität gefunden wird - die Reihenfolge oder Drehrichtung spielt also keine Rolle.

3.2 baywatch2.txt

Je weniger Zeichen gegeben sind desto schneller erfolgt die Ergebnis-Validierung. Jedoch können auch mehr mögliche Ergebnisse gefunden werden.

3.3 baywatch3.txt

Ein Beispiel für keine gegebenen Landschaftsdaten. Es erfolgt eine dementsprechende Ausgabe.

3.4 baywatch4.txt

Hier gibt es insgesamt 2 Möglichkeiten: einmal bei +300 und bei +312, die ebenfalls ausgegeben werden.

3.5 baywatch6.txt

In diesem Fall ist nur eine einziger Landschaftstyp in der Longstock-Liste markiert. Hätte man einen 128-Bit Integer könnte man dies auch durch bitweise Operationen lösen. Allerdings gilt das nur für diesen Fall und ist somit nicht allgemein anwendbar.

3.6 Sonderfall 1

Ein weiterer Sonderfall wäre der, dass keine Übereinstimmung gefunden wird. Auch dies muss dem Nutzer mitgeteilt werden und der Papagei notfalls eine Ehrenrunde drehen.

4 Quellcode

```

/* ... */

char* watch[2];
5 uint16_t count, digitcnt, *digits;

/* ... */

10 bool initBaywatch(FILE* fp) {
    char* line = NULL;
    ssize_t read;
    size_t len = 0;

    // George-Liste einlesen
15 if ((read = getline(&line, &len, fp)) != -1)
        watch[0] = strdup(line, read - 1);
    else
        return true;

    // Longstock-Liste einlesen
20 if ((read = getline(&line, &len, fp)) != -1)
        watch[1] = strdup(line, read - 1);
    else {
        free(line);
        free(watch[0]);
25         return true;
    }

    count      = read;
30 digitcnt = 0;
    digits     = (uint16_t*)malloc(sizeof(uint16_t) * read);

    // speichere zu prüfende Zeichenpositionen
35 for (len = 0; len < count; len++)
        if (isdigit(watch[1][len])) digits[digitcnt++] = len;

    digits = (uint16_t*)realloc(digits, sizeof(uint16_t) * digitcnt);

    free(line);

40 if (!digitcnt) {
        error("no biome specializations found (only '??')");
        freeBaywatch();
        return true;
45 }
    return false;
}

50 int main(int argc, const char* argv[]) {
    FILE* fp = NULL;

    // Input oder Default Datei öffnen
    if (argc > 1) tryOpen(argv[1], fp);
    if (fp == NULL && tryOpen("res/baywatch1.txt", fp)) {
55         fclose(fp);
        return 1;
    }
}

```

```

60 // Datei Einlesen & Parsen
if (initBaywatch(fp)) {
    error("initialization failed");
    fclose(fp);
    return 1;
}

65 uint16_t i, d = 0, n = 0;
printf("base:\n%s\n\n", watch[1]);

do {
    // Numerischen Buchstaben aus Longstock-Liste suchen
    while (d < count &&
        watch[0][(d + *digits) % count] != watch[1][*digits])
        d++;

    75 if (d >= count) break;

    // Mustervalidierung
    for (i = 1; i < digitcnt; i++)
        if (watch[0][(d + digits[i]) % count] != watch[1][digits[i]]) break;

    80 // Ausgabe
    if (i == digitcnt) {
        n++;
        printf(
            85 "gefunden bei +%3i:\n%s %.*s\n\n", d, watch[0] + d, d,
                watch[0]);
    }
} while (++d < count);

90 // Zusammenfassung
if (n)
    printf("%i Ergebnis%s gefunden!\n", n, n == 1 ? "" : "se");
else
    error("keine Ergebnisse gefunden!\n");

95 freeBaywatch();
fclose(fp);
return 0;
}

```