

נושאים מתקדמים בתכנות - סמ' ב' תשע"ז - תרגיל 4

בתרגיל זה עליכם לממש חלק מסויים שמתקשר למשחק הצוללות התלת-מימדי שכתבתם, אבל במנותק ממשחק הצוללות - כלומר ההגשה היא חדשה ואינה תלויה במשחק או באלגוריתם שלכם.

מציאת קבוצות במטריצה תלת-מימדית

הנח שקיימת מטריצה תלת-מימדית של איברים מסוג כלשהו. מעוניינים למפות "קבוצות" של קואורדינטות בתוך המטריצה. קבוצות של קואורדינטות שייכות לאותה קבוצה אם: (1) הן צמודות אחת לשניה באחד המימדים ו-(2) פונקציית מיפוי נתונה מחזירה עבורן את אותו מיפוי.

דוגמה ראשונה:

מטריצה שמייצגת דגימת קרקע. לכל יחידת קרקע (נקודה כלשהי במרחב) יש ערך כלשהו של צפיפות שממנו אפשר להסיק את סוג הקרקע. נתונה פונקציית מיפוי שעבור ערכי צפיפות יודעת להחזיר את סוג הקרקע. מעוניינים לקבל קבוצות של שכבות קרקע - גושי קרקע מאותו סוג שצמודים אחד לשני.

דוגמה שנייה:

מטריצה שמייצגת צוללות במרחב תלת-מימדי. לכל משבצת במרחב התלת-מימדי יש ייצוג. אפשר להחליט שכל האותיות שמייצגות צוללת כלשהי יהיו "צוללת" בין אם שלי או של היריב, כל שאר האותיות יהיו "כלום". עכשיו אפשר "לארוז" לקבוצות את כל הקואורדינטות הצמודות שמחזיקות צוללת באופן שיהיה קל לבדוק האם צוללות הן חוקיות ואם כן להחזיק את כל הקואורדינטות שלהן.

שלב ראשון (80 נקודות)

יש להגיש מימוש עבור מטריצה דו-מימדית שיתמוך ב-main הבא:

```
template<typename Groups>
void print(const Groups& all_groups) {
    for(const auto& groupType : all_groups) {
        cout << groupType.first << ":" << endl;
        for(const auto& groupOfType : groupType.second) {
            for(const auto& coord : groupOfType) {
                cout << "{ ";
                for(auto i : coord) { // was: for(int i : coord)
                    cout << i << ' ';
                }
                cout << "} ";
            }
            cout << endl;
        }
    }
}

int main() {
    Matrix2d<char> m = {{'a', 'A', 'a'}, {'B', 'a', 'B'}, {'B', 'a', 'B'}};
    auto all_groups = m.groupValues([](auto i){return islower(i)? "L": "U";});
    print(all_groups);
}
```

תוצאת ההדפסה של התכנית לעיל:

```
L:
{ 0 0 }
{ 0 2 }
{ 1 1 } { 2 1 }
U:
{ 0 1 }
{ 1 0 } { 2 0 }
{ 1 2 } { 2 2 }
```

שימו לב:

1. אנחנו נבדוק את הקוד שלכם ע"י הפעלתו וריצה על הקבוצות שמתגלות במטריצה - לא נסתמך על ההדפסה. אתם עדיין צריכים לתמוך בהדפסה כמו ב-main שלמעלה, אבל פחות חשוב הפורמט המדויק.
2. סדר הערכים בריצה בלולאה חשוב! לכן צריך להשתמש במבני נתונים שמבטיחים סדר.
3. המטריצה שלכם חייבת לספק מתודה בשם `groupValues` שמחזירה קבוצות של תאים סמוכים ששייכים לאותה קבוצה בהתאם ל-lambda שנשלחת.
4. למשתנה שנקרא `all_groups` בקוד למעלה - שהוא הערך המוחזר של הפונקציה `groupValues` יש אוסף של קבוצות שמיוצגות אצלנו באמצעות המשתנה `groupType`.
5. ל-`groupType` יש שדה `first` שמכיל את סוג הקבוצה, כאשר הערכים האפשריים הם הערכים שיכולים לחזור מפונקציית ה-lambda שנמסרה ל-`groupValues`. בנוסף יש לו שדה `second` שמחזיר את כל הקבוצות שמופּו לסוג זה.
6. המשתנה שנקרא `groupOfType` בקוד למעלה - שהוא הערך המוחזר מ-`groupType.second` הינו אוסף של כל הקבוצות מהסוג `groupType` שנמצאו, ממיינות לפי המשבצת הראשונה שבה הקבוצה נמצאה, בסדר עולה, כאשר משבצת נחשבת "קטנה" יותר אם ה-`row` שלה קטן יותר ועבור שתי משבצות עם `row` זהה, אם ה-`col` שלה קטן יותר.
7. המשתנה שנקרא `coord` בקוד למעלה - הינו מיופּי של משבצת במטריצה. עליכם להחליט כיצד ליצור מחלקה מתאימה. נדרש ש-`group` ידע להחזיר את כל ה-`coords` שלו בלולאת `range-based-for` בסדר ממוין.
8. נדרש שניתן יהיה לרוץ על `coord` בלולאת `range-based-for` ולקבל את הרכיבים של הקואורדינטה בסדר ממוין מהמימד החיצוני ביותר (במטריצה דו-מימדית: קודם מספר השורה ואח"כ מספר העמודה. במטריצה תלת-מימדית קודם מספר השכבה, אח"כ מספר שורה ובסוף מספר עמודה).

שלב שני (20 נקודות)

יש להגיש מימוש עבור מטריצה תלת-ממדית שיתמוך בקוד הבא:

```
int main() {
    Matrix3d<int> m2 = { {{1, 2, 3}, {1, 2}, {1, 2}}, {{1, 2}, {1, 2, 3, 4}} };
    auto groups = m2.groupValues([](auto i){return i%3? "!x3": "x3";});
    print(groups);
}
```

שימו לב שאין שינוי בפונ' `print` שהוצגה למעלה.

שימו לב שיכולים להיות חסרים ערכים במטריצה והם יושלמו ל-0 / `nullptr` / `false` או איתחול ברירת מחדל אחר בהתאם לסוג האיברים שבמטריצה. בדוגמה שלמעלה המטריצה תושלם ל:

```
{ {{1,2,3,0},{1,2,0,0},{1,2,0,0}}, {{1,2,0,0},{1,2,3,4},{0,0,0,0}} }
```

תוצאת ההדפסה של התכנית לעיל:

```
!x3:
{000}{001}{010}{011}{020}{021}{100}{101}{110}{111}
{113}
x3:
{002}{003}{012}{013}{022}{023}{102}{103}{112}{120}{121}{122}{123}
```

שימו לב:

כל ההערות שכתובות לגבי השלב הראשון רלבנטיות גם לשלב השני.

בונוס (6 נקודות)

במידה והמימוש של שני השלבים מתבסס על אותה מחלקה עם פרמטר template של מספר המימדים - ושתומכת גם ב-groupValues על מטריצה עם מספר כלשהו של מימדים - יינתן בונוס של ע 6 נקודות בהתאם לטיב המימוש.

דוגמת מימוש (חלקית):

```
template<class T, size_t DIMENSIONS>
class Matrix {
    //...
};

template<class T>
using Matrix2d = Matrix<T, 2>;

template<class T>
using Matrix3d = Matrix<T, 3>;
```