ĐẠI HỌC BÁCH KHOA HÀ NỘI

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----- 🙡 🕮 🙣 -----



**BÁO CÁO PROJECT I**

***Báo cáo tuần 6***

Giảng viên: **Ngô Lam Trung**

Sinh viên thực hiện: Nguyễn Quang Long

Mã số sinh viên: 20194320

**Hà Nội, năm 2021**

Nội dung đã hoàn thành:

Từ câu 127 đến câu 141

**127. Chess Triangle**

a. Đề bài

Cho bàn cờ kích thước nxm gồm 3 quân: Mã, Tượng và Xe. Hỏi có bao nhiêu cách xếp để 3 quân cờ ăn nhau theo vòng tròn?

b. Giải pháp

Chia làm 4 trường hợp bàn cờ con mà tỉ lệ kích thước bàn cờ là: 2:3, 3:3, 2:4, 3:4, ứng với mỗi tỉ lệ có các cách xếp cố định. Cộng tổng tất cả các trường hợp sẽ là kết quả

c. Code

int solution(int n, int m) {

    int twoByThree = 0;

    if((n-1) \* (m-2) > 0)

        twoByThree += (n-1) \* (m-2) \* 8;

    if((n-2) \* (m-1) > 0)

        twoByThree += (n-2) \* (m-1) \* 8;

    int threeByThree = 0;

    if((n-2) \* (m-2) > 0)

        threeByThree += (n-2) \* (m-2) \* 16;

    int twoByFour = 0;

    if((n-1) \* (m-3) > 0)

        twoByFour += (n-1) \* (m-3) \* 8;

    if((n-3) \* (m-1) > 0)

        twoByFour += (n-3) \* (m-1) \* 8;

    int threeByFour = 0;

    if((n-2) \* (m-3) > 0)

        threeByFour += (n-2) \* (m-3) \* 8;

    if((n-3) \* (m-2) > 0)

        threeByFour += (n-3) \* (m-2) \* 8;

    return twoByThree + threeByThree + twoByFour + threeByFour;

}

**129. Pawn Race**

a. Đề bài

Cho vị trí của con tốt đen và con tốt trắng trên bàn cờ vua ( tốt đen không thể đứng ở hàng đầu tiên bên đen và tốt trắng không thể đứng ở hàng đầu tiên bên trắng) Kiểm tra xem bên đen hay bên trắng thắng nếu bên đi trước là w (trắng) hoặc b (đen)

b. Giải pháp

Nếu 2 con tốt nằm cùng cột thì hòa

Nếu 2 con tốt nằm trên các cột cách nhau >= 1 ô thì tính số nước đi của mỗi quân đến hàng cuối rồi so sánh

Nếu 2 con tốt nằm trên các cột gần sát nhau thì tiến dần 2 con tốt và kiểm tra con nào sẽ ăn được con kia trước

c. Code

std::string solution(const std::string& white, const std::string& black, char toMove) {

    std::string res[3] {"draw", "white", "black"};

    const int whiteCol = white[0] - 'a';

    const int blackCol = black[0] - 'a';

    int whiteRow = white[1] - '1';

    int blackRow = black[1] - '1';

    if (whiteCol == blackCol && blackRow > whiteRow)

        return res[0];

    if (std::abs(whiteCol - blackCol) > 1 || blackRow <= whiteRow) {

        const int whiteToMove = 7 - (whiteRow) - (whiteRow == 1 ? 1 : 0);

        const int blackToMove = (blackRow) - (blackRow == 6 ? 1 : 0);

        return whiteToMove + (toMove == 'b' ? 1 : 0) <= blackToMove ? res[1] : res[2];

    }

    while (true) {

        if (blackRow - whiteRow == 1)

            break;

        switch (toMove) {

        case 'w':

            whiteRow += (whiteRow != 1 || blackRow - (whiteRow + 2) == 1) ? 1 : 2;

            break;

        case 'b':

            blackRow -= (blackRow != 6 || (blackRow - 2) - whiteRow == 1) ? 1 : 2;

            break;

        }

        if (blackRow == 0 || whiteRow == 7)

            break;

        toMove = toMove == 'w' ? 'b' : 'w';

    }

    return res[toMove == 'w' ? 1 : 2];

}

**130. Valid Time**

a. Đề bài

Kiểm tra xem 1 xâu có phải là 1 xâu chỉ thời gian chính xác (trong đồng hồ 24h) không?

b. Giải pháp

Tách xâu ra giờ và phút, nếu giờ <24 và phút <60 thì thỏa mãn, ngược lại thì không.

c. Code

bool solution(string time) {

    int hour = (int)(time[0]-'0')\*10 + (int)(time[1]-'0');

    int minute = (int)(time[3]-'0')\*10 + (int)(time[4]-'0');

    if(hour<24&&minute<60) return true;

    return false;

}

**131. Video Part**

a. Đề bài

Cho 2 xâu biểu diễn thời gian đã xem và tổng độ dài của 1 video, biểu diễn thời gian đã xem bằng phân số.

Ví dụ

For part = "02:20:00" and total = "07:00:00", the output should be  
solution(part, total) = [1, 3].

b. Giải pháp

Chuyển thời gian về giây rồi chia cho ước chung lớn nhất của 2 số

c. Code

int seconds (string total){

     int hour = (int)(total[0]-'0')\*10+ (int)(total[1]-'0');

    int minute = (int)(total[3]-'0')\*10+ (int)(total[4]-'0');

    int second = (int)(total[6]-'0')\*10+ (int)(total[7]-'0');

    return hour\*3600 + minute\*60 + second;

}

vector<int> solution(string part, string total) {

   int tol = seconds(total);

   int part1 = seconds(part);

   int g = gcd(part1, tol);

   return {part1/g,tol/g};

}

**132. Day of Week**

a. Đề bài

Cho 1 xâu gồm ngày tháng năm, tìm xem sau bao nhiêu năm nữa thì có cùng thứ, ngày, tháng với năm đã cho

b. Giải pháp

Nếu ngày tháng đó là 29/02 thì sau mỗi 4 năm mới có 1 năm có ngày 29/02, các ngày khác có sau mỗi năm

Sau mỗi 1 năm không nhuận thì thứ của ngày đang xét lệch đi 1 và năm nhuận lệch đi 2, như vậy sau khi lệch đi đủ 7 ngày thì quay trở về thứ ban đầu

Lưu ý, các năm chia hết cho 100 và ko chia hết cho 400 không phải là năm nhuận

c. Code

function solution(birthdayDate) {

    let date = new Date(birthdayDate);

    let year = date.getFullYear();

    let month = date.getMonth();

    let day = date.getDate();

    let solution = date.getDay();

    let result = 0;

    while (true) {

        result += (29 == day && 1 == month) ? 4 : 1;

        if (0 == (year + result) % 100 && 29 == day && 1 == month) {

            continue;

        }

        if (solution == (new Date(year + result, month, day)).getDay()) {

            return result;

        }

    }

    return result;

}

**133. Curious Clock**

a. Đề bài

Cho 2 xâu chỉ thời gian ( giờ - phút – ngày – tháng – năm), xâu thứ nhất chỉ thời điểm đồng hồ bị quay ngược lại, tính xem thời điểm ở xâu thứ 2 trên chiếc đồng hồ đó là bao nhiêu?

b. Giải pháp

Tính khoảng thời gian từ lúc bắt đầu quay ngược đến thời gian cần tính rồi lấy thời điểm bắt đầu quay ngược lùi về từng đó thời gian

c. Code

function solution(someTime, leavingTime) {

    let some = new Date(someTime);

    let leave = new Date(leavingTime);

    let sub = leave - some;

    let res = new Date(some - sub);

    return res.toJSON().substr(0, 16).replace('T', ' ');

}

**134. New Year Celebrations**

a. Đề bài

Một người đi máy bay về hướng Tây, vì vậy múi giờ giảm dần. Tính xem người này có thể đón năm mới bao nhiêu lần (khi đi qua mỗi múi giờ kém hơn thì thời gian lùi 1 giờ) với thời điểm xuất phát và thời gian đi qua mỗi múi giờ cho trước?

b. Giải pháp

Chuyển thời gian về phút, thời gian xuất phát ở múi thứ i là t\_i, và thời điểm giao thừa là 24x60 = 1440, thời điểm đến múi giờ tiếp theo là t\_i+1. Nếu t\_i<=1440 và t\_i+1 >=1440 thì khi qua múi giờ đó người đó có thể lại đón giao thừa 1 lần. Và nếu thời điểm hạ cánh <=1440 thì người đó có thể đón giao thừa thêm 1 lần nữa.

c. Code

int solution(std::string t, std::vector<int> c)

{

    int hour = stoi(t.substr(0, 2));

    int minute = stoi(t.substr(3, 2));

    int time = hour \* 60 + minute;

    const int newyear = 24 \* 60;

    int count = 0;

    if(time==0)  time = newyear;

    int rootTime = time;

    for(int i =0; i<c.size();i++){

        int nextTime = rootTime + c[i] - 60\*i;

        time -= (i==0?0:60);

        if(time <= newyear && nextTime >=newyear) count++;

        time = nextTime;

    }

       time -=60;

    if(time <= newyear) count++;

    return count;

}

**135. Regular Months**

a. Đề bài

Cho currMonth là tháng-năm có ngày đầu tiên trong tháng là thứ 2, tìm xem tháng-năm tiếp theo có ngày đầu tiên trong tháng cũng là thứ 2?

b. Giải pháp

Sử dụng hàm dateTime, nếu ngày đầu tiên trong tháng không phải là thứ 2 thì tăng tháng thêm 1.

c. Code

function solution(currMonth) {

  currMonth = currMonth.split("-").join("-01-");

  var date = new Date();

  date.setTime(Date.parse(currMonth));

  date.setMonth(date.getMonth() + 1);

  while (date.getDay() !== 1) {

    date.setMonth(date.getMonth() + 1);

  }

  return `${("0" + (date.getMonth() + 1)).substr(-2)}-${date.getFullYear()}`;

}

**136. Missed Classes**

a. Đề bài

Cho mảng dayOfTheWeek chỉ các thứ trong tuần có lớp học, biến year chỉ năm học và mảng holidays gồm các ngày nghỉ lễ từ 1/9 năm nay đến 31/5 năm sau. Tính xem có bao nhiêu ngày nghỉ lễ trùng với ngày có buổi học.

b. Giải pháp

Duyệt lần lượt các ngày nghỉ lễ và tính thứ của ngày đó, nếu nằm trong mảng dayOfTheWeek thì tăng thêm 1.

c. Code

function solution(year, daysOfTheWeek, holidays) {

  var date = new Date();

  daysOfTheWeek = daysOfTheWeek.map((d) => d % 7);

  var makeUpDays = 0;

  for (var i = 0; i < holidays.length; i++) {

    if (Number(holidays[i].split("-")[0]) >= 9) {

      date.setTime(Date.parse(holidays[i] + "-" + year));

    } else {

      date.setTime(Date.parse(holidays[i] + "-" + (year + 1)));

    }

    if (daysOfTheWeek.includes(date.getDay())) {

      makeUpDays++;

    }

  }

  return makeUpDays;

}

**137. Holiday**

a. Đề bài

Cho các biến x, weekday, month, year, tính xem vào thứ weekday trong tuần thứ x của của tháng month năm year là ngày bao nhiêu.

b. Giải pháp

Ngày đầu tiên của tuần thứ x là 7x-6, tăng dần thêm 1 ngày, nếu trùng với thời gian input thì trả về ngày đó, nếu nhảy sang tháng khác thì trả về -1.

c. Code

function solution(x, weekDay, month, yearNumber) {

    let weekDayNum = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"].indexOf(weekDay);

    let monthNum = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"].indexOf(month);

    let date = new Date(yearNumber, monthNum, 7\*x-6);

    while (date.getDay() != weekDayNum) {

        date = new Date(date.getTime() + 24\*60\*60\*1000);

    }

    if (date.getMonth() != monthNum) {

            return -1;

        }

    return date.getDate();

}

**138. Is Sentence Correct?**

a. Đề bài

1 sentence được xem là đúng khi bắt đầu bằng chữ cái in hoa, kết thúc bằng ! hoặc . hoặc ? và . ? ! không xuất hiện ở giữa sentence. Cho 1 sentence, kiểm tra xem nó có đúng không?

b. Giải pháp

Sử dụng regex

c. Code

bool solution(std::string sentence) {

  std::regex regex("^[A-Z][^.?!]\*[.?!]$");

  return std::regex\_match(sentence, regex);

}

**139. Replace All Digits RegExp**

a. Đề bài

Cho 1 xâu, thay thế tất cả các chữ số bằng ký tự #

b. Giải pháp

Sử dụng hàm regex\_replace với regex [0-9]

c. Code

std::string solution(std::string input) {

  return regex\_replace(input,regex("[0-9]"),"#");

}

**140. Swap Adjacent Words**

a. Đề bài

Cho 1 sentence, đổi chỗ 2 từ liên tiếp đôi một với nhau

b. Giải pháp

Sử dụng regex\_replace

c. Code

std::string solution(std::string s) {

  return std::regex\_replace(s, std::regex("([a-zA-Z]+) ([a-zA-Z]+)"), "$2 $1");

}

**141. N-th Number**

a. Đề bài

Cho 1 xâu s gồm ít nhất n số (số ở đây là 1 chuỗi các ký tự số liên tiếp), các số này có thể chứa các ký tự ‘0’ ở đầu nhưng phải có ít nhất 1 số khác 0. Hãy tìm số thứ n và trả về xâu không có các số 0 ở đầu

b. Giải pháp

Sử dụng regex để tìm các xâu là số thứ n rồi dùng regex\_match

c. Code

std::string solution(std::string s, int n) {

  std::regex regex("(?:[^1-9]\*([0-9]+)){"+to\_string(n)+"}.\*");

  std::smatch match;

  std::regex\_match(s, match, regex);

  return match[1];

}