

## 変更履歴

t\_station.same\_flg は int から char に変更空文字列が重複駅なし。文字列が含まれていると重複駅ありとなる。  
文字列は駅名表示時、他の駅と区別のつかないように記した文字列(「(東)」とか、「(海)」とか)  
この変更の為に、駅名表示時、クエリ結果列を、ltrim(name)+same\_flg とすることができる

jctflg は cityflg, spe\_route へ移す

ダイクストラの高速化のため

また会社線乗り換えを考慮すること

新横浜から京都までの運賃計算算出方法は？

次の経路は、片道乗車券として、○発券できる、×発券できない。

○岐阜→名古屋→(新幹線)→米原→岐阜

×名古屋→(新幹線)→米原→(在来線)→名古屋

※ 米原-名古屋間(両駅除く)の在来線駅発着または合流・分岐の場合のみ許される

# UI(iPhone)

---

151 条(分岐駅)

経路指定時、自動計算(経路は変更しないが適用情報を保持して開示可能にする)

70 条

経路指定時、自動計算(経路は変更しないが適用情報を保持して開示可能にする)

新幹線と在来線乗換

経路指定時、自動計算(経路は変更しないが適用情報を保持して開示可能にする)

69 条

経路指定時、自動計算(経路は変更しないが適用情報を保持して開示可能にする)

157 条 2 項

86 条／87 条

経路指定時は適用しない。以下ボタンをクリックすることにより適用／非適用が切り替わる

「北九州市内発」／「単駅発」、と「北九州市内着」／「単駅着」の 2 つのボタンが有効化される

経路指定時はすべて指定経路の営業キロと運賃を表示

69 条、70 条は自動適用して計算

86 条、87 条も自動適用して計算(但し、非適用ボタンがある)

**単駅指定** ボタンを有効化

**都区市内発** ボタン有効化

**都区市内着** ボタン有効化

**114 条** ボタンを有効化

**157 条** ボタンを有効化

**最短** ボタンを有効化

---

同名駅は漢字上で話で読みが異なっても同名とはしない

---

C++ライブラリ、

SQLITE3 C++ Wrapper class を使用(The Code Project 版のものを使用: 改変自由、商用使用可)

発駅 TextField 選択ボタン>>

経由

運賃・有効期間表示エリア

---

---

---

---

## データ数、制限値

厳密には正確ではないが設計には有用な値

項目	数	
路線数	202	
駅数	4589	
分岐駅数	310	
t_lines	4965	(旧) 新幹線、分岐駅数分 +

### 制限値

項目	数
路線・駅数(t_lines レコード数)	16383=14 ビット(0x3fff)

※ 制限値を越えるようだと設計の見直しが要となる。

# データ構造

## データ関連図

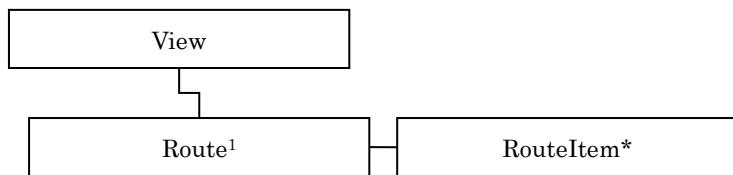


図 1

RouteItem

通常ルート(ルート typeL or typeI)

0	0x0000	長津田
1	横浜線	東神奈川
2	東海道	川崎
3	南武	立川
4	中央東	高尾

発駅

着駅

閉じたルート(ルート typeO or typeP)

0	0xffff	長津田
1	横浜線	東神奈川
2	東海道	川崎
3	南武	立川
4	中央東	八王子
5	横浜線	長津田

発駅

着駅

## QueryResultSet オブジェクト

### 説明

SQLite クエリー結果を得るイテレータオブジェクト

### クラス変数

```
string qryStationsFromLineAndPrefect  
string qryStationsFromStation  
string qryStationFromLine  
string qryLineFromPrefect  
string qryLineFromCompany
```

これらは、QueryResultSet オブジェクトをアプリケーションレイヤから使用する場合に利用  
内部で使用する場合、直接 SQL ステートメント文字列を指定する

### インスタンス変数

```

    - string sql_statement;

コンストラクタ

    QueryResultSet(const string& sql);
    QueryResultSet();

```

#### メソッド

```

vector<string>* exec(const string& sql, const vector<string>&
param);
vector<string>* exec(const vector<string>& param);

```

結果は、配列の配列(2次元配列)で返します。エラーの場合、nullを返します。

## RouteItem オブジェクト

### 説明

RouteItem オブジェクトはルート(経路)を定義するオブジェクトで、開始駅から着駅まで動的配列またはリスト)で保持する。

### クラス変数

なし

### インスタンス変数

```

int16 stationId; //発着駅(1つめは発駅)
int16 lineId; // 路線(1つめは以下※)
                // 0の場合、営業キロの算出はおこなわない(旅客営業規則 70 条処理)
int16 optStationId; //「新幹線分岐特例」時の表示上分岐駅(例:恵那→三島時、
                    lineId='新幹線',stationId='金山'時の「名古屋」)
int16 flag; //spe_route と cityflag, jctflagなどを集約(使用するもの)
            bit0-15: cityflag.b0-15
                    都区市内着発適用時は都区市内番号が設定される
            bit24-31: spe_route.b31-24

```

※ 開始駅の lineId には、以下の値が格納される

- ビット 0: 発駅が都区市内
- ビット 1: 着駅が都区市内
- ビット 2: 発駅が山手線内
- ビット 3: 着駅が山手線内
- ビット 4: 都区市内通過(発駅)
- ビット 5: 都区市内通過(着駅)

ビット 6: 100km/200km 以下の都区市内発着で発駅を単駅指定可能  
 ビット 7: 100km/200km 以下の都区市内発着で着駅を単駅指定可能

※ 86/87 条適用時、ビット 6=ON の場合、「発駅を単駅に指定」と表示し、指定された場合、適用フラグ値 1(ビット 0=ON)で呼び出すことにより、発駅を単駅とし、着駅を都区市内適用にして結果を表示する(このとき、この適用フラグのビット 6 は Off にし、ビット 7 を ON にする。  
 ビット 7=ON の場合、適用フラグ値 2(ビット 1=ON)で呼び出すことにより、着駅を単駅とし、発駅を都区市内適用にして結果を表示する(このとき、この適用フラグのビット 7 は Off にし、ビット 6 を ON にする。

ビット 8: 発駅は、大阪・新大阪(ビット 0~3 優先)  
 ビット 9: 着駅は、大阪・新大阪(ビット 0~3 優先)

0: I 字ルート  
 -1: これ以上先へはいけないルート(○字、P 字)

```
int16 alternate_stationId; // 実際の分岐駅(例:「名古屋(金山)」の名古屋)、
                           0 の場合は通常
```

※ イタリック体は未実装

#### メソッド

```
void stationName(string* station, string* station_line); // 駅名
                                                               と駅の所属する'/'で区切った複数の路線名を返す。
void lineName(string* line); // 路線名を返す。発駅の場合は空文字を返す。
```

## Route オブジェクト

### クラス変数

なし

### インスタンス変数

- int passingJunction\_[ (MAX\_JCT + 7) / 8]; // ルートリスト
- vector<RouteItem> route\_items;
- int16 y\_branch\_station; // b: calc\_km(分岐駅) (新旭川)  
 (最も最近の特例分岐駅)
- int16 y\_branch\_line; // c: sales\_km(本線) (宗谷)  
 (最も最近の特例分岐駅)

※ 最終追加 route\_items の spe\_route.BSRJCTSP が OFF のときは、  
 y\_branch\_station、y\_branch\_line は、0 に初期化する(Must で

はないはず)

#### メソッド

```
void setStartStation(int stationId) // 始発駅の追加  
void clear(); // ルートリストを消去  
int add(int lineId, int stationId1, int stationId2); // ルートの追加  
void removeTail(); // ルートの最終項目を削除  
int itemAt(int index, const RouteItem& route_items); // RouteItemを得る  
int size(); // ルート数を得る  
void calcFare(FARE_RESULT* result); // 運賃計算
```

#### イテレータメソッド

RouteIterator\* iterator(); 開始のイテレータを返す。RouteIterator オブジェクトはシングルトンとして静的ポインタを返す(delete 不要)

## RouteIterator オブジェクト

#### 説明

構築時にイテレータインデックスを初期化する

#### クラス変数

なし

#### インスタンス変数

- int current\_index;
- vector<RouteItem>\* route\_items;

#### メソッド

```
bool hasNext(); // つづきの有無を返す  
RouteItem* next(); // カレント RouteItem を返す
```

#### 使用方法:

```
RouteIterator *routeIterator = route.iterator();  
RouteItem* routeItem;  
while (routeItem = routeIterator->next()) {  
    routeItem->xxx  
    :  
}
```

## void Route::setStartStation(int stationId) 始発駅の追加

ルートリストの開始駅を設定する。既に設定されている場合は上書きし、途中まで格納されていたルートリストがあればそれもすべて削除される。

## ルートリスト (int passing\_junction\_[(MAX\_JCT + 7)/8])

発駅と着駅を除き分岐駅のみに通過マークをセットする。分岐マークは 40 バイトの配列にビット単位で設定する。

分岐駅は t\_jct テーブルに stationId と Id として変換できるようにしている。

分岐駅の総数は 311 で、MAX\_JCT には 320 と定義される。 $(MAX\_JCT + 7) / 8 = 40$

## 乗車経路マーク

```
Int Route.add(int lineId, int stationId1, int stationId2)
```

乗車経路をマークする。のちの片道乗車券判定時の重複経路になつてないかの判定に使用する。片道乗車券とは、一度通った駅を通らない一筆書きの経路である。例として、L の字、I の字、O の字は許されるが、Q の字や、9 の字は許されない。但し 6 の字は許される。

路線の分岐駅 && 営業キロ  $\leq$  駅 1 && 駅 2  $\leq$  営業キロ order by sales\_km ||  
路線の分岐駅 && 営業キロ  $\leq$  駅 2 && 駅 1  $\leq$  営業キロ order by sales\_km  
desc

で列挙される駅の分岐 ID を足跡配列の該当ビットオフセットを ON にする。

既に ON になつていたら重複エラーとなる。正し、発駅が ON の場合は良しとし、着駅が ON の場合は最後とする(最終駅警告)(ルート終了。閉じたルート。いわゆるルート typeO または、typeP)

足跡配列数は、 $(MAX\_JCT + 7) / 8$

MAX\_JCT は、311 なので、余裕を見て 320 とある。320 ビット=40 バイトの配列

クエリ一文を以下にしめす

パラメータは、?1=lineId, ?2=station1, ?3=station2

```
select id, t.name from t_lines l join t_station t on t.rowid=l.station_id  
join t_jct j on j.station_id=l.station_id where  
line_id=?1 and jctflg!=0 and  
case when  
(select sales_km from t_lines where line_id=?1 and station_id=?3) <  
(select sales_km from t_lines where line_id=?1 and station_id=?2) then  
((sales_km>=(select sales_km from t_lines where line_id=?1 and  
station_id=?3)) and  
(sales_km<=(select sales_km from t_lines where line_id=?1 and  
station_id=?2))) ) else  
(((sales_km<=(select sales_km from t_lines where line_id=?1 and  
station_id=?3)) and  
(sales_km>=(select sales_km from t_lines where line_id=?1 and  
station_id=?2)))) ) end  
order by
```

```

case when
(select sales_km from t_lines where line_id=?1 and station_id=?3) <
(select sales_km from t_lines where line_id=?1 and station_id=?2) then
sales_km
end asc,
case when
(select sales_km from t_lines where line_id=?1 and station_id=?3) >
(select sales_km from t_lines where line_id=?1 and station_id=?2) then
sales_km
end desc;

```

※ 以前は上り用と下り用の二つのクエリを発行していたが一回でOKのようにした(ただし同じ条件文が複数回あり、冗長なので、プリコンパイル済みクエリーが使用するべき。

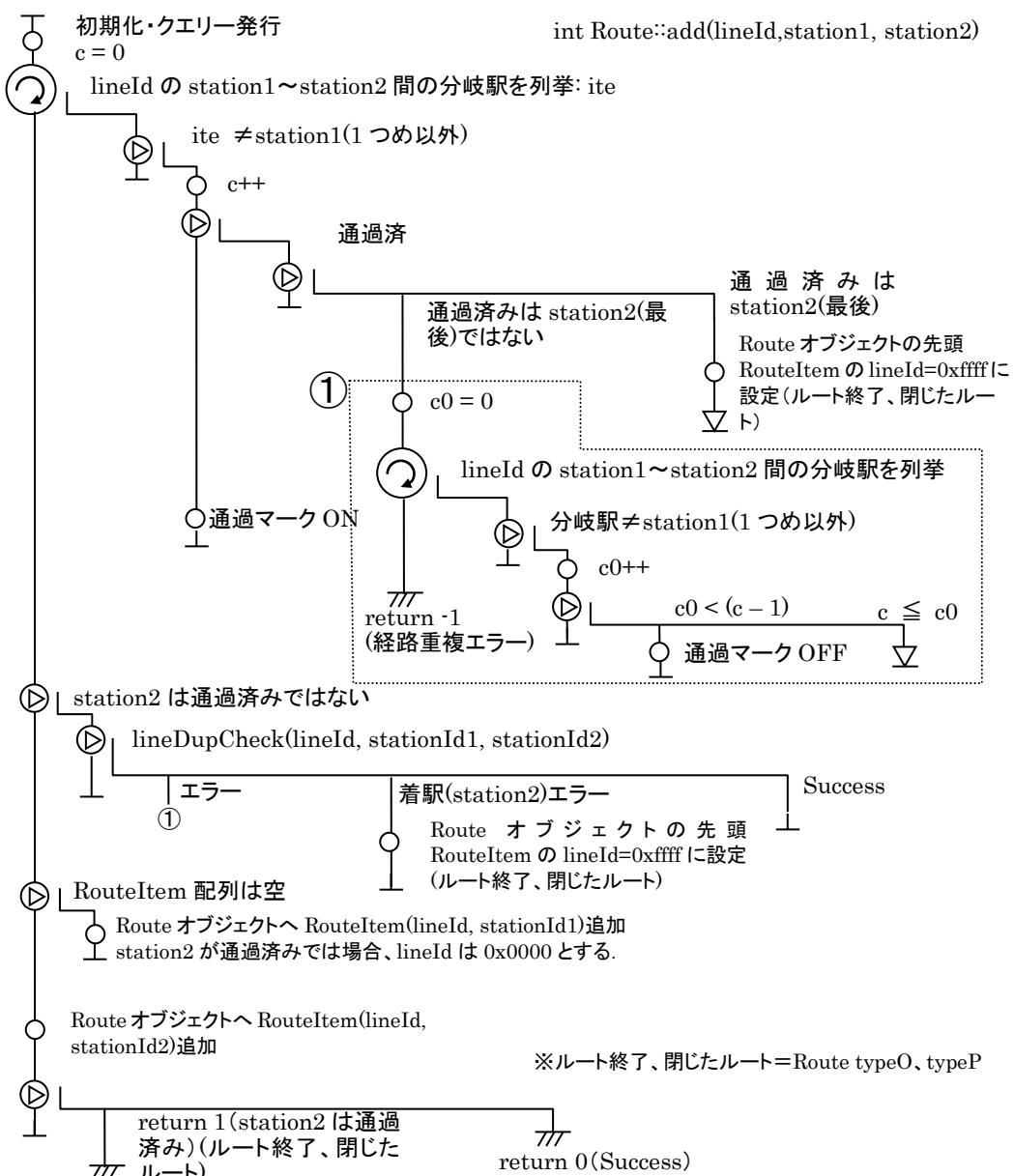


図 2

※ フローは細かすぎてコーディングできないのでみなおし

## 関数値

- 0: 成功
  - 1: 経路重複
  - 1: 経路重複(stationId2, typeO, P)  
MSB(if 0 or 1)=1(0x8000\_0000 or 0x8000\_0001) : 特例によるルート変更  
あり。→ 表示時、全アイテムをリフレッシュして一覧更新する必要がある。それ  
以外は、最終アイテムの追加のみの対応で可能
- 

```

o junction
. junction(dont check yet)
g goal
s start
left station1
right station2
(g) 着指定
<g> 着指定／終了駅どちらも可(の筈)
0 正常終了
1 正常終了(最後)(つぎの removeTail で last item の通過マスクを Off しない)
2 正常終了(最後)(つぎの removeTail で last item の通過マスクを Off する)
-1 異常終了
///////////
stationId2 == endStationId の場合は 2 とする(但し 1 の場合もある)
呼び出し側は endStationId のみが終わりとは限らない(着指定の場合もあるので)

```

```

rc == 0
last_flag = 0
return 1
rc == 1
last_flag = 1
return 0
rc == 2
last_flag = 0
return 0
rc < 0
last_flag = 0
return -1

```

```

removeTail():
if ((i != 0) || (last_flag == 0)) {
    mask off
}

```

		alloff: y = alloff / L = last on / x = 途中 on	rc	al l off	
a	-----o-----o	通常(東京-富士-甲府)	0	y	
d	-----o-----g	通常(東京-富士-<g>国母)	0	y	
g	g -----o-----o	東京-富士-(g)甲府 八王子-甲府-富士-東神奈川 ※ g == s もテスト時考慮	0	y	

		町田・橋本・茅ヶ崎・東神奈川・町田			
f	-----o---s	八王子・甲府・富士・蒲田 町田・東神奈川・川崎・立川・八王子・町田 町田・八王子・立川・川崎・東神奈川・町田	2	y	
v	-----o---s-g	町田・橋本・茅ヶ崎・東神奈川・古淵 町田・東神奈川・川崎・立川・八王子・成瀬 町田・八王子・立川・川崎・東神奈川・古淵  inStation(stationId2, startStationId, (nextStationId))    町田・八王子間に古淵があるか? 町田・八王子間に町田があるか?にならないように startStationId!=stationId2 条件判定後 inStation(startStationId, stationId1, stationId2) 東神奈川・古淵間に町田はあるか? 東神奈川・町田間に町田はあるか?にならないように startStationId!=stationId2 条件判定後	-1	y	start
l	-----o-g--o	東京・富士-((g)身延)-甲府	-1	y	end
p	---g-o---o	八王子・甲府・富士-((g)二宮)-品川 八王子・甲府・富士-((g)二宮)-有楽町	-1	y	end
t	g ---o-----o	茅ヶ崎-(g 鶴見)-東京 茅ヶ崎-(g 鶴見)-蒲田	-1	y	end
w	-----o---g-s	町田・橋本・茅ヶ崎・東神奈川-(g 成瀬)-町田 町田・東神奈川・川崎・立川・八王子-(g 古淵)-町田 町田・八王子・立川・川崎・東神奈川-(g 成瀬)-町田	-1	y	
b	-----o---x	八王子・東神奈川・茅ヶ崎・橋本	1	L	
j	s -----o---x	八王子・甲府・富士・東神奈川・八王子 ※ g == s モテスト時考慮	1	L	
h	g -----o---x	八王子・東神奈川・茅ヶ崎-(g)橋本 八王子・東神奈川・茅ヶ崎-g 橋本 新秋津・新松戸・日暮里・南浦和 ※着駅は着席指定、事前指定の 2 種で試すこと	1	L	
x	s -----o---g-.	橋本・茅ヶ崎・東神奈川-(g 成瀬)-橋本 東神奈川・川崎・立川・八王子-(g 町田)-東神奈川 八王子・立川・川崎・東神奈川-(g 町田)-八王子	-1	L	
k	s ----x---.	八街・佐倉・成田・我孫子・新松戸・西船橋-(佐倉 x)-八街	-1	L	
e	-----x---g	町田・橋本・茅ヶ崎・東神奈川-(橋本<x>)-(g)片倉	-1	L	
r	-----o-s-x	町田・橋本・茅ヶ崎・東神奈川-(町田)-橋本	-1	L	
u	g ---o-----x	八王子・甲府・富士・東神奈川-(g 橋本)-八王子	-1	L	end
m	-----o-g--x	八王子・東神奈川・茅ヶ崎-(g)海老名・橋本 八王子・橋本・茅ヶ崎・東神奈川-(g)町田・橋本	-1	L	
c	-----x----.	横浜・東神奈川・橋本・茅ヶ崎-(横浜 x)-(東神奈川 x)-川崎 八王子・橋本・茅ヶ崎・東神奈川-(橋本 x)-(八王子 x) 渋谷・新宿・御茶ノ水 府中本町-(南浦和)-新松戸・日暮里-(南浦和 x)-高崎 町田・橋本・茅ヶ崎・東神奈川-(橋本 x)-八王子	-1	x	
i	g -----x----.	町田・橋本・茅ヶ崎・東神奈川-(橋本 x)-<g>八王子	-1	x	
n	---g-o---x ---g-x----.	府中本町-(南浦和)-新松戸・日暮里-(南浦和 x)-(g)浦和-大宮 府中本町-(南浦和)-新松戸・日暮里-(g)西日暮里-大宮 稲城長沼・府中本町-(南浦和)-新松戸・日暮里-(g)西日暮里-大宮 東川口・西国分寺・立川・拝島・高麗川・大宮-(g)浦和-日暮里 ※着駅は着席指定、事前指定の 2 種で試すこと	-1	x	end
o	-----x-g--.	町田・橋本・茅ヶ崎・東神奈川-(橋本<x>)-(g)片倉-八王子	-1	x	end
q	-----o-s-o ---s-o---o ---s-o---x	南浦和・武蔵浦和-大宮-上野 橋本・茅ヶ崎・川崎・立川・八王子・東神奈川 (町田・橋本・茅ヶ崎・川崎・立川・八王子・東神奈川) (片倉・橋本・茅ヶ崎・川崎・立川・八王子・東神奈川) 川崎・武蔵小杉・鶴見-(川崎 x)-品川	1	x	

		※ s は分岐駅以外あり得ない			
s -----x-s--		新秋津・南浦和・東京・立川・府中本町・武藏浦和/新松戸	1	x	start
s ---s-x----		保土ヶ谷・東神奈川・橋本・茅ヶ崎・品川	1	x	

end:

endStationId in the last between stationId1 and stationId2

start:

startStationId in the last between stationId1 and stationId2

//////////

分岐駅-同一線内(分岐駅含まない)-分岐駅

分岐駅-同一線内(分岐駅含む)-分岐駅

非分岐駅-同一線内(分岐駅含まない)-分岐駅

非分岐駅-同一線内(分岐駅含む)-分岐駅

非分岐駅-同一線内(分岐駅含まない)-非分岐駅

非分岐駅-同一線内(分岐駅含む)-非分岐駅

分岐駅-同一線内(分岐駅含まない)-非分岐駅

分岐駅-同一線内(分岐駅含む)-非分岐駅

type X 武藏小杉

type 6 八王子-東神奈川-茅ヶ崎-橋本

type 9 橋本-茅ヶ崎-東神奈川-八王子

type O 橋本-東神奈川-川崎-立川-八王子-橋本

type O 長津田-東神奈川-川崎-立川-八王子-長津田

**Int Route::lineDupCheck(lineId, stationId1, stationId2)**

長津田-東神奈川-川崎-立川-八王子-十日市場みたいなの(八王子-十日市場は NG)を検知する。

十日市場-東神奈川間は NG

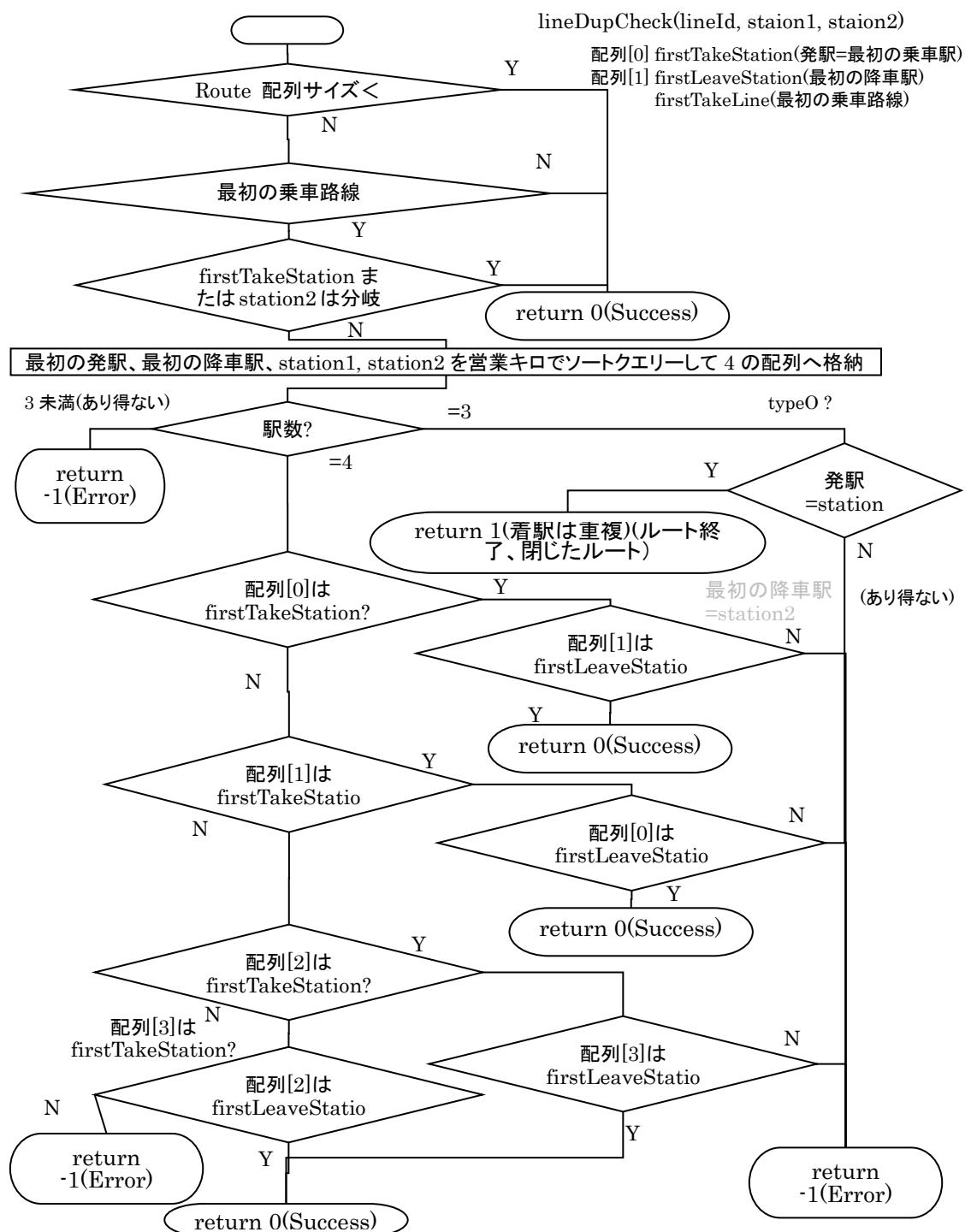
八王子-長津田間は OK

橋本の次-長津田間は閉塞区間なので終着

駅 1 が最初の路線と同一路線に限り、最初の発駅～降車駅、駅 1、駅 2 を営業キロでソートし、駅 1、駅 2 が発駅～降車駅内にないこと(2 本の線が重なり合わないこと)。

→新幹線、在来線の考慮も必要になる例は？

発駅に限ってチェックしているのは、途中分岐駅は分岐駅チェックでひっかかるので分岐駅以外というのは発駅と着駅以外にないので。



3

上記フローは不採用

以下にクエリーを高等化した簡易処理版をしめす。

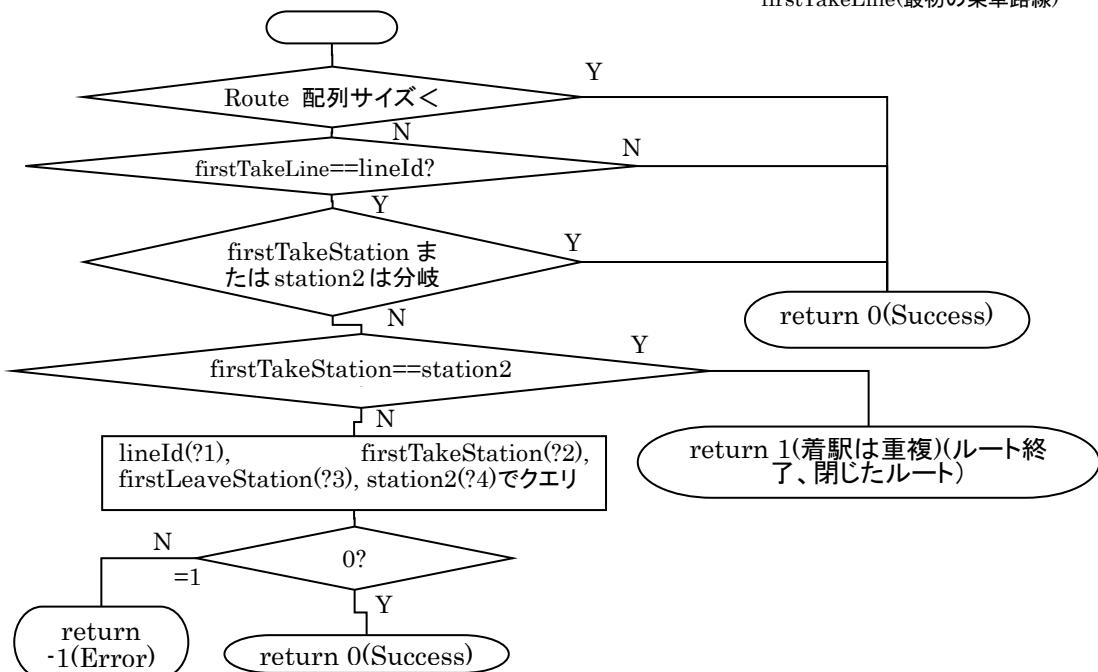
以下のクエリーで、横浜線(?1)長津田(?2)～東神奈川(?3)間に駅(?4)が含まれていないかをチェックします。

```
select count(*) from (
  select station_id
  from t_lines
  where line_id=?1
  and station_id
  in (select station_id
       from t_lines
       where line_id=?1
       and ((sales_km>=(select sales_km
                           from t_lines
                           where line_id=?1
                           and station_id=?2)
          and (sales_km<=(select sales_km
                           from t_lines
                           where line_id=?1
                           and station_id=?3)))
         or (sales_km<=(select sales_km
                           from t_lines
                           where line_id=?1
                           and station_id=?2)
          and (sales_km>=(select sales_km
                           from t_lines
                           where line_id=?1
                           and station_id=?3))))))
) where station_id=?4;
```

含まれていれば、1を返し、含まれていなければ0を返す。

含まれている駅(?4)を増やすには、クエリ文に「or station\_id=?5」を追加するのみである

lineDupCheck(lineId, staion1, staion2)  
配列[0] firstTakeStation(発駅=最初の乗車駅)  
配列[1] firstLeaveStation(最初の降車駅)  
firstTakeLine(最初の乗車路線)



```
void Route::removeTail();
```

ルート配列の末尾を削除する。

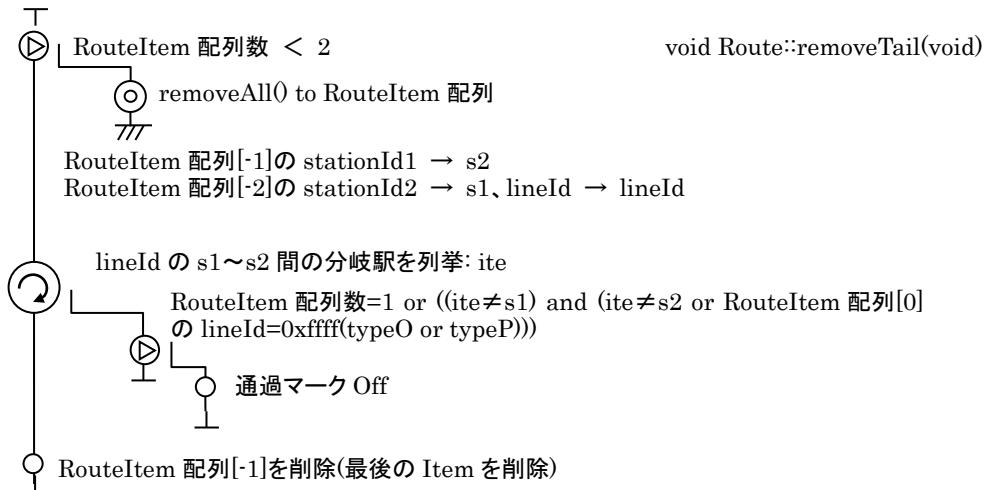


図 4

### ● 分岐特例駅の取得

a, d から、b, c の取得(t\_lines テーブルより)

ex.) spe\_route.BSRJCTSP=ON の“石北線-旭川”のレコードの実分岐駅“c 新旭川”、通過列車  
分岐駅の所属路線“b を得る

#### <入力>

a: 分岐路線(石北)

d: 乗換駅(旭川)

#### <出力>

b: sales\_km(本線)(宗谷) → route.y\_branch\_line へ保持

c: calc\_km(分岐駅)(新旭川) → route.y\_branch\_station へ保持

※ spe\_route.BSRJCTSP が OFF のときは、y\_branch\_station、  
y\_branch\_line は、0 に初期化される(Must ではないはず)

### ● 路線Iの a 駅～b 駅間に駅 c はあるか？

```
int isInStation(int lineId, int station1, int station2)  
return 0=no / 1=あり
```

➔ 「lineDupCheck()」で使用したクエリーをつかう

### ● 路線の a 駅→b 駅は、上りか下りか？

```
int dirLine(int lineId, int station1, int station2)  
return 0=station1=station2 / 1=上り/-1=下り
```

以下のクエリーは、路線?1 の駅?2→駅?3 への乗車が上りなら 1 を下りなら 0 を返す。

(駅 1=駅 2 の場合、下り(0)を返す)

```
select case when  
  ((select sales_km from t_lines where line_id=?1 and station_id=?2) -  
   (select sales_km from t_lines where line_id=?1 and station_id=?3)) < 0  
then 0 else 1 end;
```

### ● 新幹線の判定

line\_id のある値以上を新幹線とする。ある値は、起動初期化時、t\_line テーブルを id 順に列挙し、最初に「新幹線」を含むレコードの id を保持することにより決定される。#define(final int) 定義する。

データ作成時、新幹線は t\_line の後半になるようとする。

将来、新規路線が追加になった場合は、すべてのデータを再構築する。

## ■重複路線

路線名称と区間が重複する場合の対応

新幹線区間のダイクストラ検索時など

例

	東京	→		東京	→		東京
東海道	熱海		東海道	熱海		東海道	名古屋
新幹線	静岡		東海道	静岡			
東海道	掛川		東海道	掛川			
新幹線	名古屋		東海道	名古屋			

新幹線、熱海-静岡は、東海道線に置き換えるべき

では、

	新横浜	→		新横浜
新幹線	新富士(東)		新幹線	小田原
			東海道	三島
			新幹線	新富士(東)

新幹線、新横浜-熱海は、新横浜-小田原、東海道線-

	東京	→		東京
長野北陸新幹線	長野		東北線	大宮
			高崎線	高崎
			長野北陸新幹線	長野

新幹線を指定しているのであれば、新幹線のままの表記の方が誤解することがない。

知っているユーザであれば、路線を指定するはずである。

したがって指定したまんまとする。

ダイクストラで新幹線と入り混じる件についてはアルゴリズムの優先度で対処できる問題であり、これは新幹線優先とする。

## ■基定 151 条（分岐駅を通過する列車に乗車する場合の特

例）対応

### 分岐駅を通過する列車に乗車する場合の特例区間一覧

	a 分岐路線	c 分岐駅	d 乗換駅	b 本線
	釧網線	東釧路	釧路	根室線
	石北線	新旭川	旭川	宗谷線
	千歳線	白石	札幌	函館線
	札沼線	桑園	札幌	函館線

	千歳線	沼ノ端	苫小牧	室蘭線
	江差線	五稜郭	函館	函館線
	海峡線	中小国	蟹田	津軽線
	五能線	川部	弘前	奥羽線
	男鹿線	追分	秋田	奥羽線
	仙山線	羽前千歳	山形	奥羽線
	左沢線	北山形	山形	奥羽線
	水郡線	安積永盛	郡山	東北線
	陸羽西線	余目	酒田	羽越線
	上越線	宮内	長岡	信越線@
	鳥山線	宝積寺	宇都宮	東北線@
	両毛線	新前橋	高崎	上越線@
	八高線	倉賀野	高崎	高崎線@
	横浜線	東神奈川	横浜	東海道線
	中央東線	神田	東京	東北線*
	中央東線	代々木	新宿	山手線*
	中央西線	塩尻	松本	篠ノ井線
	中央西線	金山	名古屋	東海道線
	七尾線	津幡	金沢	北陸線
	越美北線	越前花堂	福井	北陸線
	湖西線	近江塙津	敦賀	北陸線
	湖西線	山科	京都	東海道線
		新大阪	大阪	
		大阪	新大阪	
	福知山線	尼崎	大阪	東海道線
	赤穂線	東岡山	岡山	山陽線
	伯備線	倉敷	岡山	山陽線
	芸備線	備中神代	新見	伯備線
	伯備線	伯耆大山	米子	山陰線
	瀬戸大橋線	宇多津	丸亀	予讃線
	土讃線	多度津	丸亀	予讃線
	鳴門線	池谷	勝瑞	高徳線
	徳島線	佐古	徳島	高徳線
	徳島線	佃	阿波池田	土讃線
	予讃線(向井原-内子)	向井原	伊予市	予讃線
	予土線	北宇和島	宇和島	予讃線
	吳線	※海田市	広島	山陽線
	可部線	横川	広島	山陽線
	山陰線	幡生	下関	山陽線
	日豊線	西小倉	小倉	鹿児島線
	篠栗線	吉塚	博多	鹿児島線
	唐津線	久保田	佐賀	長崎線
	日田彦山線	城野	小倉	日豊線
	日田彦山線	夜明	日田	久大線
	三角線	宇土	熊本	鹿児島線
	宮崎空港線	田吉	南宮崎	日南線

※ 三原以遠 — 広島間新幹線利用の場合を含む

※ 大都市近郊区間は除外(してはならない???)

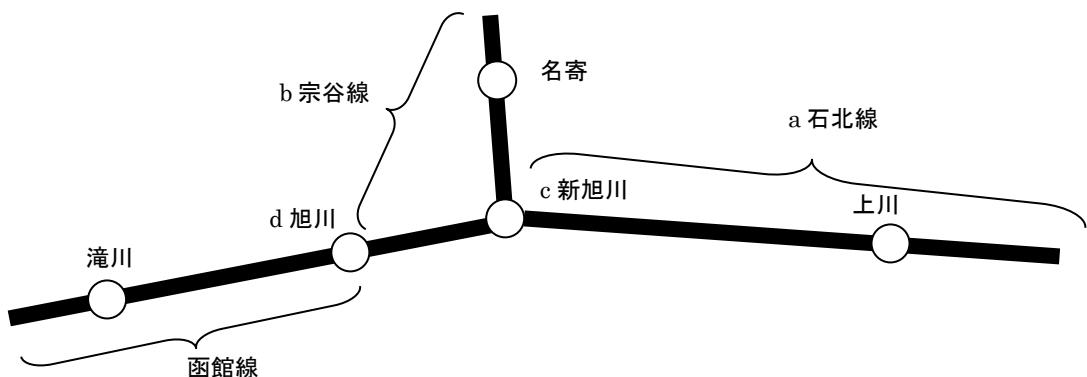
## データ定義と実装

t\_lines spe\_route.BSRJCTSP

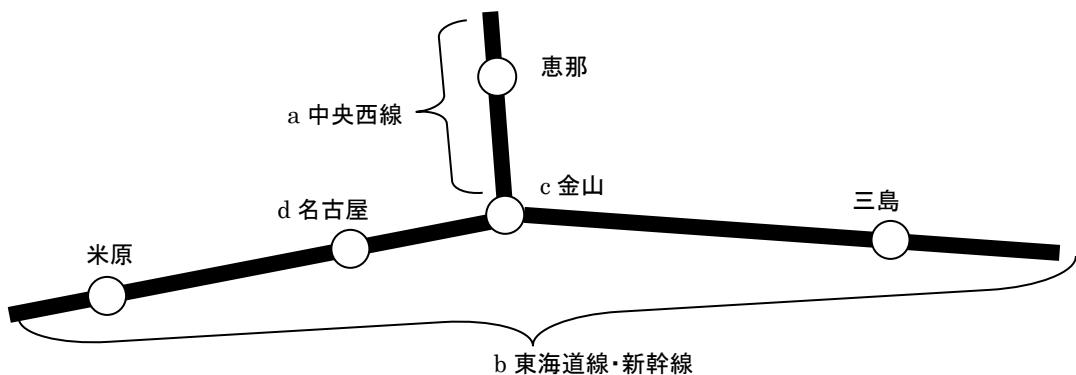
下図の例において、路線 a の d は spe\_route.BSRJCTSP=1 とする

この場合、t\_lines.calc\_km の下位 16 ビットは、c(t.station.rowid)、calc\_km の上位 16 ビットは、b(t\_line.rowid)が格納されているものとする。また、lflg の分岐駅属性 BCJCTFLG も ON となる

### 在来線のみ



### 新幹線を含む例



石北線レコードの旭川は spe\_route が ON である

### 実装

駅選択後、route.add()内で、チェックをおこなう。

<rule1> 路線の着駅が spe\_route のとき  
着駅→c に置換  
b-d を追加

<rule2> 直前着駅選択時に spe\_route 置換(<rule1>適用)があった  
if 路線 is 新幹線 && 新幹線.発駅 d-着駅間に c があるか?  
b-d を削除

```

        (最初 d だった) c を d(c) に置換
else if 路線==b
    b-d を削除(最後の route_item を削除<rule2a>
else
    そのまま(指定ルート追加)

```

※ 最初の路線選択時はなにもスルー

```

<rule3> 路線-発駅(選択路線の直前着駅)が spe_route のとき
    if 前路線 is 新幹線 && 新幹線-(発駅-駅)間に(c)駅がある
        "d(c)" 内に置換内部では(c)に
    else if 前路線!=(b)
        前路線の前に(b-c)を挿入 <rule3a>
    else
        発駅(直前着駅)を(c)に置換 <rule3b>

```

route.add()

Rule1 適用?

Rule1 適用フラグ(Rule2)?

Rule3 適用?

この順番とする(たぶん順不同で全チェックで問題なし)。

複合あり得るか? →あり得る

a,d から、b, c の取得(t\_lines テーブルより)

ex.) spe\_route.BSRJCTSP=ON の“石北線-旭川”のレコードの実分岐駅”c 新旭川”、通過列車  
分岐駅の所属路線”b を得る

<入力>

a: 分岐路線(石北)

d: 乗換駅(旭川)

<出力>

b: HI\_WORD(calc\_km)(本線)(宗谷) → route.y\_branch\_line へ保持

c: LO\_WORD(calc\_km)(分岐駅)(新旭川) → route.y\_branch\_station へ保持

※ spe\_route.BSRJCTSP が OFF のときは、y\_branch\_station 、  
y\_branch\_line は、0 に初期化される(Must ではないはず)

以下、例に沿って説明する。

### Ex.1 上川-滝川／恵那-米原

<rule1><rule2>

	上川	→		上川
石北 a	旭川 d		石北 a	新旭川 c
函館線	滝川		宗谷 b	旭川 d

旭川を選択	→	函館線(選択)
		富良野線
		宗谷

「石北-旭川」を追加時、石北線の旭川が spe\_route.BSRJCTSP=ON なので<rule1>適用として、  
 「石北-旭川」から、y\_branch\_station, y\_branch\_line を取得し、route インスタンス変数に  
 保持する。旭川 d を新旭川 c に置換する。石北 a-旭川 d の旭川 d の spe\_route.BSRJCTSP は、  
**rule1 適用フラグをあらわすものとし、RouteItem の flag の BSRJCTSP bit も ON にする。**

仮に最後に「宗谷 b-旭川 d」(y\_branch\_line-lineId)を追加する。

つぎに「函館線-滝川」を追加時、route\_items[last].flag.BSRJCTSP=1(rule1 適用フラグ  
 が ON)なので<rule2>のチェックとして、函館線(lineId) != b なので“仮”は仮ではなくそのまま終  
 了。

	恵那	→		恵那			恵那
中央西	名古屋		中央西 a	金山 c		中央西	金山 c
新幹線	米原		東海道 b	名古屋 d		東海道	米原

(参考:これでも OK)

「中央西-名古屋」追加時、<rule1>で d 名古屋を c 金山に置換、b 東海道-d 名古屋追加

「新幹線-米原」追加時、<rule2>で新幹線名古屋 d～米原間に金山 c がないので、そのまま新幹線  
 -米原を追加

## Ex.2 上川-名寄／恵那-三島

<rule1><rule2a>

	上川	→		上川			上川
石北	旭川		石北	新旭川		石北	新旭川
宗谷	名寄		宗谷	旭川		宗谷	名寄

旭川を選択 →	函館
	富良野
	宗谷(選択)

「石北-旭川」を追加時、石北線の旭川が spe\_route.BSRJCTSP=ON なので<rule1>適用として、  
 「石北 a-旭川 d」から、y\_branch\_station, y\_branch\_line を取得し、route インスタンス変  
 数に保持する。旭川 d(lineId)を新旭川 c(y\_branch\_station)に置換する。石北 a-旭川 d の  
 旭川 d の spe\_route.BSRJCTSP は、rule1 適用フラグをあらわすものとし、

RouteItem.flag.BSRJCTSP=ON にする。

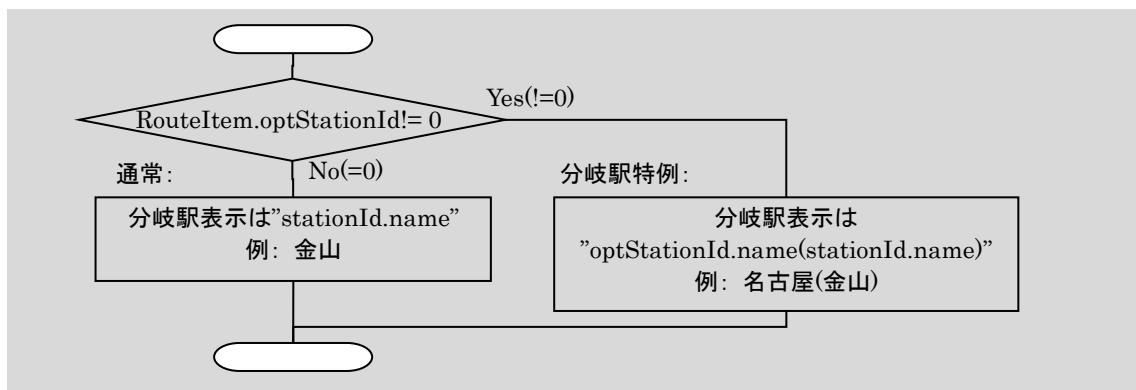
仮に最後に「宗谷 b-旭川 d」(y\_branch\_line-lineId)を追加する。

つぎに「宗谷 b-名寄」を追加時、route\_items[last].flag.BSRJCTSP=1(rule1 適用フラグが ON)なので <rule2> のチェックとして、宗谷 (lineId)=b(y\_branch\_line) なので route\_items[last] (仮に追加した「宗谷 b-旭川 d」(=y\_branch\_line-y\_branch\_station))を削除してから、lineId, stationId(宗谷 b-名寄 d)を追加する。

### 恵那-三島(!新幹線分岐特例)

恵那		恵那		恵那	
中央西 a	名古屋 d	→	中央西 a	金山 c	→
新幹線	三島		東海道 b	名古屋 d	
			新幹線	三島	

「中央西 a-名古屋 d」追加時、<rule1>で名古屋 d → 金山 c 変換、東海道 b-名古屋 d 追加  
 「新幹線-三島」追加時、<rule2>で新幹線なので、新幹線、名古屋 d-三島間に金山 c があるので、  
 「新幹線-三島」を追加するまえに、最終アイテムである「東海道 b-名古屋 d」を削除、金山 c を“名古屋 d (金山 c)”に置換。内部的には、RouteItem.optStationId= 名古屋 d にする  
 (RouteItem.stationId は金山 c のまま)ことで、表示時、以下のようになる。



### Ex.3 滝川-上川／米原-恵那

<rule3a>

滝川		滝川	
函館線	旭川	→	函館線
石北	上川		宗谷
			新旭川

「石北-上川」追加時、石北線乗車駅(t\_lines.lineId + route\_items[last].stationId) (=石北-旭川) が spe\_route の場合(route.add())の引数 stationId.BSRJCTFLG=1 であるかで決定される)、<rule3> 適用となり、「石北 - 旭川 (lineId-route\_items[last].stationId)」から、y\_branch\_station,

`y_branch_line` を取得し、`route` インスタンス変数に保持する。前路線 (`route_items[last].lineId`) 函館線 != 宗谷 (b) (`y_branch_line`) のので、宗谷 (`y_branch_line`) - 新旭川 (`y_branch_station`) (b-c) を追加してから、石北 (`lineId`) - 上川 (`stationId`) を追加する。

#### 米原-恵那

	米原	→		米原		米原
新幹線 b	名古屋 d		新幹線	名古屋 d		東海道
中央西 a	恵那		東海道 b	金山 c		中央西
			中央西	恵那		(参考:これでもOK)

「中央西-恵那」追加時、`<rule3>`で、前路線は新幹線で米原-名古屋間に金山がなく、前路線:新幹線 != 東海道 b で、東海道 b-金山 c を追加してから「中央西-恵那」を追加`<rule3b>`

#### 米原-恵那(在来線経由)

	米原	→		米原
東海道 b	名古屋 d		東海道 b	金山 c
中央西 a	恵那		中央西 a	恵那

`<rule3b>`が適用され名古屋を金山に置換

#### Ex.4 名寄-上川／三島-恵那

`<rule3b>`

	名寄	→		名寄
宗谷	旭川		宗谷	新旭川
石北	上川		石北	上川

「石北-上川」追加時、石北線乗車駅(`t_lines.lineId + route_items[last].stationId`) (=石北-旭川) が `spe_route` の場合 (`route.add()` の引数 `stationId.BSRJCTFLG=1` であるかで決定される)、`<rule3>` 適用となり、「石北-旭川 (`lineId-route_items[last].stationId`)」から、`y_branch_station`, `y_branch_line` を取得し、`route` インスタンス変数に保持する。前路線 (`route_items[last].lineId`) 宗谷 = 宗谷 (b) (`y_branch_line`) のので、旭川 (`route_items[last].stationId`) を新旭川 (c) (`y_branch_station`) に置き換え。  
`<rule3b>`

#### 三島-恵那(!新幹線分岐特例)

	三島	→		三島	→		三島
新幹線	名古屋 d		新幹線	名古屋 d		新幹線	名古屋(金山)
中央西 a	恵那		東海道 b	金山 c		中央西 a	恵那
			中央西 a	恵那			

※名古屋-金山間は双方の駅を含め途中下車できません。

「中央西-恵那」追加時、<rule3>で、前路線は新幹線で、三島-名古屋間に金山 c があるので、新幹線降車駅(名古屋 d)を金山 c にし、optStationId を名古屋 d にする。(表示は「名古屋(金山)」となります)

その後に「中央西-恵那」を追加

最後の条件表示時にも以下の分を追加する。

「※名古屋-金山間は双方の駅を含め途中下車できません。」

### Ex.5 滝川-名寄／米原-三島

	滝川
函館線	旭川
宗谷	名寄

	米原
新幹線	名古屋
東海道	三島

	米原
東海道	名古屋
新幹線	三島

	米原
新幹線	三島

	米原
東海道	三島

通常ルート：ルール適用無し

### Ex.6 名寄-滝川／三島-米原

	名寄
宗谷	旭川
函館	滝川

通常ルート：ルール適用無し

旭川で終わりの場合(分岐選択と着駅選択が事前に明確なため問題ない)

### Ex.7 上川-旭川／恵那-名古屋

	上川
石北	旭川

は、<rule1><rule2> で

	上川
石北	新旭川
宗谷	旭川

	恵那	<rule1>		恵那
中央西	名古屋	<rule2>	中央西	金山
		で、→	東海道	名古屋

### Ex.8 旭川から上川の例(名古屋から恵那の例)

<rule3a>

	旭川			旭川
石北	上川		宗谷	新旭川
			石北	上川

	名古屋			名古屋 d
中央西	恵那		東海道 b	金山 c
			中央西 a	恵那

「石北-上川」追加時、石北線乗車駅(t\_lines.lineId + route\_items[last].stationId) (=石北-旭川) が spe\_route の場合(route.add())の引数 stationId.BSRJCTFLG=1 であるかで決定される)、<rule3> 適用となり、「石北 - 旭川 (lineId-route\_items[last].stationId)」から、y\_branch\_station, y\_branch\_line を取得し、route インスタンス変数に保持する。前路線(route\_items[last].lineId) NULL!= 宗谷 (b) (y\_branch\_line) なので、宗谷 (y\_branch\_line)-新旭川 (y\_branch\_station) (b-c) を追加してから、石北 (lineId)-上川 (stationId) を追加する。

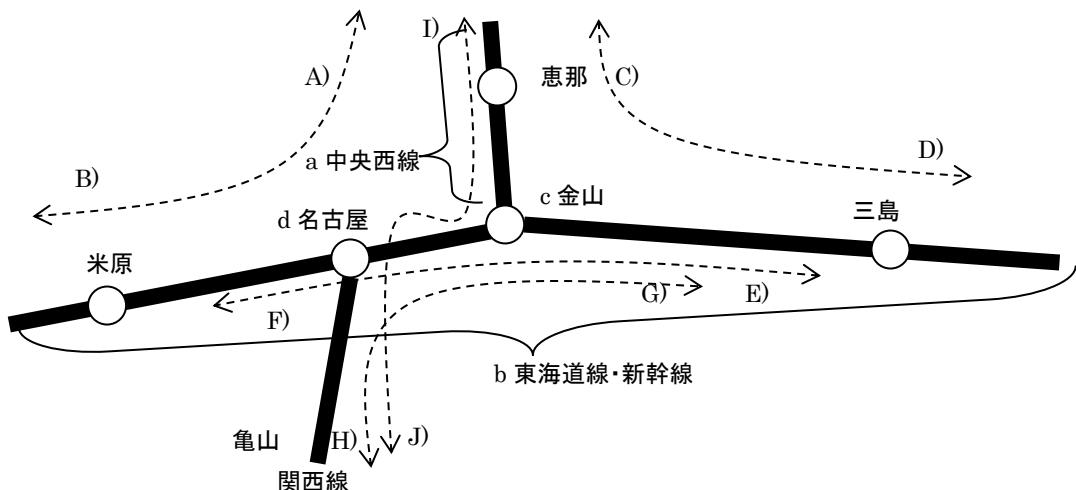
### Ex.9 上川-富良野／恵那-龜山

	上川			上川
石北	旭川	→	石北	新旭川 c
富良野線	富良野		宗谷 b	旭川 d
			富良野線	富良野

	恵那	→		恵那
中央西	名古屋		中央西 a	金山 c
関西	亀山		東海道 b	名古屋 d
			関西	亀山

「中央西-名古屋」追加時、<rule1>で名古屋→金山変換、東海道-名古屋追加

「関西-亀山」追加時、<rule2>で亀山!=東海道なのでそのまま関西-亀山を追加



t\_lines の中央西線 (line\_id):名古屋 (station\_id) は、(lflg & (1<<31)) !=0 で、東海道線 (HI\_WORD(calc\_km)), 金山 (LO\_WORD(calc\_km)) と定義されている。

分岐特例の変換は、経路追加時: add() 関数内でおこなう。

凡例:

白抜きは、分岐特例駅で路線.駅が分岐特例駅(路線 a.駅 c)

太字は、変更部分、グレー背景は挿入経路をしめす。

## ➤ 段差型

駅 2 ≠ 金山、LastIndex 路線=b

A-1)

	大阪
東海道線	名古屋
中央西線	恵那

	大阪
東海道線	金山
中央西線	恵那

駅 1 置換

路線 LastIndex (=b) の LastIndex-1 (大阪) - 駅 1 (LastIndex:名古屋) (d) 間に c (金山) がある場合、c (金山) から d (名古屋) を分岐マーク off する

駅 1 (d) → c (金山) に置換して継続

C-1)

	横浜
東海道線	名古屋
中央西線	恵那

	横浜
東海道線	金山
中央西線	恵那

駅 1 置換

A-1 と同様

駅 2 ≠ 金山、LastIndex 路線 ≠ b

A-0)

	d 名古屋
a 中央西線	恵那

	名古屋
b 東海道線	c 金山
中央西線	恵那

挿入

路線 b (東海道線)、駅 1=LastIndex (=d:名古屋)、駅 2=c (金山) として、add() 呼出、

路線 a (=元引数の路線:中央西線) )、駅 1=c (金山)、駅 2=元引数の駅 2 として継続

レコード挿入時 lflg.b31=On

I)

	亀山
関西線	名古屋
中央西線	恵那

	亀山
関西線	名古屋
東海道線	金山
中央西線	恵那

挿入

A-0 と同様

駅 2 ≠ 金山、LastIndex 路線=b を並行在来線とする新幹線

A-2)

	新大阪
新幹線	名古屋
中央西線	恵那

	新大阪
新幹線	名古屋
東海道線	金山
中央西線	恵那

挿入

A-0 と同様

駅 2 ≠ 金山、LastIndex 路線=b を並行在来線とする新幹線

C-2)

	新横浜
新幹線	名古屋
中央西線	恵那

	新横浜
新幹線	名古屋 → 三河安城
東海道線	金山
中央西線	恵那

非表示

挿入

LastIndex 路線(新幹線) の LastIndex-1 の駅(新横浜) ~ LastIndex (=駅 1:名古屋) 間に  
c (金山) は存在するか?

存在する場合、c(金山)から d(名古屋)を分岐マーク Off にし、LastIndex (=駅 1:名古屋)を、LastIndex 路線(新幹線)の LastIndex (=駅 1:名古屋)から LastIndex-1 駅(新横浜)方向の隣の在来線接続駅(三河安城)に入れ替え、路線 b(東海道線)、駅 1 を(三河安城)-駅 2 を c(金山)で、add()呼出(エラーは無し)、元引数の路線 (=a:中央西線)、駅 1 を駅 2(c:金山)、駅 2 を元引数の駅 2(恵那)にし、継続。1flg.b31=On とする。

駅 2=金山

路線(a:中央西線)を、b:東海道線にして継続

E-0)

	大阪
東海道線	名古屋
中央西線	金山

	大阪
東海道線	名古屋
東海道線	金山

路線置換

E-1)

	大阪
東海道線	名古屋
中央西線	金山
東海道線	横浜

	大阪
東海道線	名古屋
東海道線	金山
東海道線	横浜

路線置換

E-2)

	新大阪
新幹線	名古屋
中央西線	金山
東海道線	横浜

	新大阪
新幹線	名古屋
東海道線	金山
東海道線	横浜

路線置換

G)

	亀山
関西線	名古屋
中央西線	金山
東海道線	横浜

	亀山
関西線	名古屋
東海道線	金山
東海道線	横浜

路線置換

G-3)

	名古屋
中央西線	金山
東海道線	横浜

	名古屋
東海道線	金山
東海道線	横浜

路線置換

G-2)

	名古屋
中央西線	金山

	名古屋
東海道線	金山

路線置換

G-4)

	名古屋
中央西線	金山
東海道線	大阪

	名古屋
東海道線	金山
東海道線	大阪

路線置換

-> 経路重複エラー(金山-大阪間に名古屋あり)

### ➤ 水平型

水平型は、中央西線-名古屋着で、発は、金山か、金山以外の駅のみである。

前者の場合、中央西線を東海道線に置換。後者の場合、駅2を金山に置換し、東海道線-名古屋を追加する。

F-0)

	金山
中央西線	名古屋

	金山
東海道線	名古屋

中央西線を東海道線に置換

F-1)

	横浜
東海道線	金山
中央西線	名古屋
東海道線	大阪

	横浜
東海道線	金山
東海道線	名古屋
東海道線	大阪

F0 の後、  
RemoveDupRoute()

水平型検知して変換(中央西線を東海道線に置換)後、フラグを立てて、次回分岐チェック後の経路配列追加時に RemoveDupRoute() する。

F-2)

	金山
中央西線	名古屋
東海道線	大阪

	金山
東海道線	名古屋
東海道線	大阪

F0 の後、  
RemoveDupRoute()

E-10)

	金山
中央西線	名古屋
東海道線	横浜

	金山
東海道線	名古屋
東海道線	横浜

F0 の後、  
RemoveDupRoute()

-> 経路重複エラーとなる(名古屋-横浜間に金山が既通)

H)

	横浜
東海道線	金山
中央西線	名古屋
関西線	亀山

	横浜
東海道線	金山
東海道線	名古屋
関西線	亀山

F0 の後、  
RemoveDupRoute()

E-11)

	大阪
東海道線	金山
中央西線	名古屋
東海道線	横浜

	大阪
東海道線	金山
東海道線	名古屋
東海道線	横浜

F0 の後、  
RemoveDupRoute()

-> 経路重複エラー(大阪-名古屋間に金山が既通)

E-12)

	大阪
東海道線	金山
中央西線	名古屋
東海道線	大垣

	大阪
東海道線	金山
東海道線	名古屋
東海道線	大垣

F0 の後、  
RemoveDupRoute()

-> 経路重複エラー(大阪-名古屋間に金山が既通)

E-13)

	横浜
東海道線	金山
中央西線	名古屋
東海道線	静岡

	横浜
東海道線	金山
東海道線	名古屋
東海道線	静岡

F0 の後、  
RemoveDupRoute()

-> 経路重複エラー(名古屋-静岡間に金山が既通)

E-14)

	横浜
東海道線	金山
中央西線	名古屋
東海道線	金山

	横浜
東海道線	金山
東海道線	名古屋
東海道線	金山

F0 の後、  
RemoveDupRoute()

-> 経路重複エラーではないけど。複乗なのでエラーにする

F-3)

	金山
中央西線	名古屋
新幹線	新大阪

	金山
東海道線	名古屋
新幹線	新大阪

F0 の後、  
RemoveDupRoute()

F-3)

	金山
中央西線	名古屋
新幹線	新横浜

	金山
東海道線	名古屋
新幹線	新横浜

F0 の後、  
RemoveDupRoute()

-> 経路重複エラーとなる

J-0)

	恵那
中央西線	名古屋

	恵那
中央西線	金山
東海道線	名古屋

駅 2 を金山に置換し、東海道線-  
名古屋を追加

add() 内で、水平型検知したら、駅 2 を金山に置換して add() を再帰呼び出しし、line\_id=を東海  
道線、駅 1 を金山、駅 2 を名古屋にして水平型検知フラグを立て、継続処理

B-1)

	恵那
中央西線	名古屋
東海道線	大阪

	恵那
中央西線	金山
東海道線	名古屋→大阪

J0 の後、  
RemoveDupRoute()

add() 内で水平型検知フラグが ON で、直前路線=挿入予定路線の場合、重複経路チェックをおこない、

経路追加時、駅 2 (大阪) を LastIndex-1 の駅 (名古屋) に置き換える (フラグは Off)。

D-1)

	恵那
中央西線	名古屋
東海道線	横浜

	恵那
中央西線	金山
東海道線	名古屋→横浜

J0 の後、  
RemoveDupRoute()

B-2)

	恵那
中央西線	名古屋
新幹線	新大阪

	恵那
中央西線	金山
東海道線	名古屋
新幹線	新大阪

J0 の後、  
RemoveDupRoute()

水平型検知フラグは ON であるが、直前路線≠挿入予定路線なので、そのまま継続(フラグは Off)。  
挿入予定路線が直前路線の並行新幹線の場合、新幹線(挿入予定路線)の駅 1 (名古屋) ~ 駅 2 (新大阪) 間に LastIndex-1 (挿入前の) の駅 (金山) が存在するか? しない場合は (B-2)、そのまま、  
存在した場合、駅 1 を新幹線の駅 2 方面の駅 1 の隣の在来線接続駅 (三河安城) に置換し (D-2)、  
処理を継続

挿入予定路線が直前路線の並行新幹線でない場合もそのまま処理を継続 (J)

D-2)

	恵那
中央西線	名古屋
新幹線	新横浜

	恵那
中央西線	金山
東海道線	名古屋 ↓ 三河安城
新幹線	新横浜

J0 の後、  
RemoveDupRoute()

新横浜の隣の在来線接続駅または、  
名古屋の手前の在来線接続駅を得る

B-2) の説明通り

J)

	恵那
中央西線	名古屋
関西線	亀山

	恵那
中央西線	金山
東海道線	名古屋
関西線	亀山

J0 の後、  
RemoveDupRoute()

B-2) の説明通り

### 連続パターン

	松本
中央西線	名古屋
新幹線	新横浜

	松本
篠ノ井線	塩尻
中央西線	金山
東海道線	三河安城
新幹線	新横浜

駅  $2 \neq c$  で、LastIndex 路線  $\neq b$  ので、 $b-c$  (篠ノ井線-塩尻) 挿入、b31 を ON にした、中央西線-名古屋をそのまま追加処理に入るが、水平型適用でもあるので

	新横浜
新幹線	名古屋
中央西線	松本
篠ノ井線	篠ノ井

	新横浜
新幹線	三河安城
東海道線	金山
中央西線	塩尻
篠ノ井線	松本
篠ノ井線	篠ノ井

	新横浜
新幹線	名古屋
中央西線	松本
中央東線	東京
東北線	大宮

	新横浜
新幹線	三河安城
東海道線	金山
中央西線	松本
中央東線	神田
東北線	大宮

	新横浜
新幹線	三河安城
東海道線	金山
中央西線	塩尻
中央東線	神田
東北線	大宮

	左沢
左沢線	山形
仙山線	仙台

	左沢
左沢線	北山形
奥羽線	羽前千歳
仙山線	仙台





以上をまとめると、

#### 段差型

```
line1 == sales_km(東海道線)
station2 != calc_km(金山)
置換 station1 <- calc_km(名古屋->金山) // A-1, C-1
```

```

else
    置換 line2 <- sales_km(東海道線) // E-1)
    RemoveDupRoute()
line1 == sales_km の新幹線?
station2 != calc_km(金山)
    新幹線 station0 -> station1 間に calc_km があるか?
    置換 station1:新幹線の着駅 station0 方面の station1 の在来線接続の隣駅(三河安城)
    挿入 line2 sales_km=calc_km(東海道線-金山) // C-2
else
    挿入 line2 sales_km=calc_km(東海道線-金山) // A-2
else
    置換 line2 <- sales_km(東海道線) // E-2)
    RemoveDupRoute()

line1 == 上記以外(関西線)
station2 != calc_km(金山)
    挿入 line2 sales_km=calc_km(東海道線-金山) // I
else
    置換 line2 <- sales_km(東海道線) // G
    RemoveDupRoute()

line1 == 0 (開始)
station2 != calc_km(金山)
    挿入 line2 sales_km=calc_km(東海道線-金山) // A-0 (A-2と同様)
else
    置換 line2 <- sales_km(東海道線) // G-2 (Gと同様でも可)

```

### 水平型

```

line2 == sales_km(東海道線)
station0 != calc_km(金山)
    置換 station1 <- calc_km(名古屋->金山) // B-1, D-1
else
    置換 line1 <- sales_km(東海道線) // F-1, F-2
    RemoveDupRoute(line0)

line2 == sales_km の新幹線?
    挿入 line2 sales_km=calc_km(東海道線-station1(名古屋)) // B-2, D-2
    置換 station1(名古屋) <- calc_km(金山)

    新幹線 station1(名古屋) -> station2(新横浜+新大阪) 間に calc_km(金山) があるか?
    または、分岐フラグが ON か? でも良い
    置換 station1:新幹線の着駅 station2 方面の station1 の在来線接続の隣駅(三河安城)

line2 == 上記以外(関西線)
station0 != calc_km(金山)
    挿入 line2 sales_km=station1(東海道線-名古屋) // J
    置換 station1 <- calc_km(金山)
else
    置換 line1 <- sales_km(東海道線) // H
    RemoveDupRoute(line0)

```

### 新幹線の隣の在来線接続駅を返すクエリー

```

select l.station_id from t_lines where line_id=?1 and (lflg&(1<<22))=0 and
sales_km<
(select max(sales_km) from t_lines where line_id=?1 and (station_id=?2 or

```

```
station_id=?3)) order by sales_km desc limit(1);
```

?1 東海道新幹線

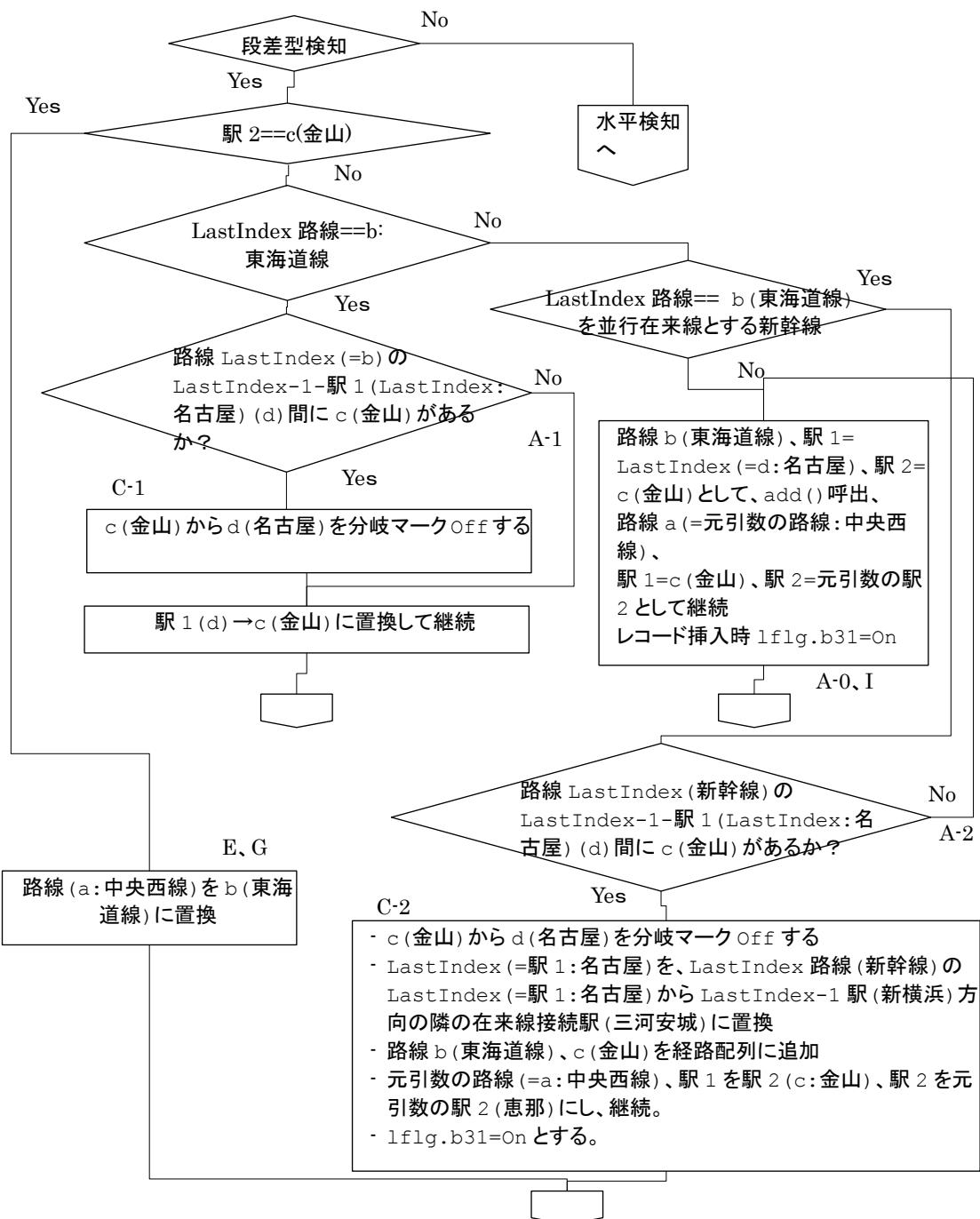
?2 新横浜

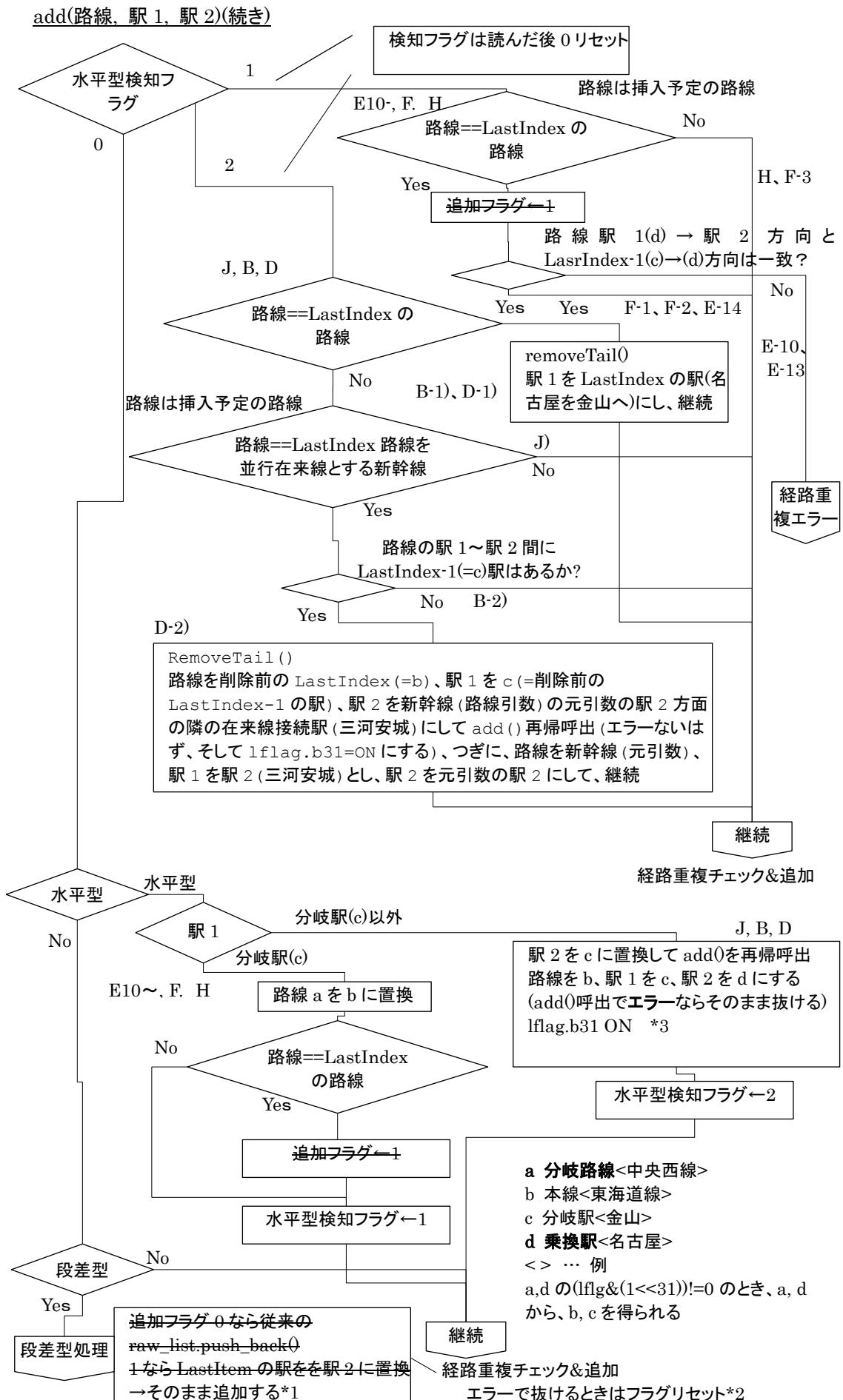
?3 名古屋

?2 が新大阪や京都にすると、米原が返される

?2 が新横浜や東京にすると、三河安城が返される

### add(路線、駅 1、駅 2)





\*1 あとに RemoveTail()されることを考慮すると、そのまま追加。

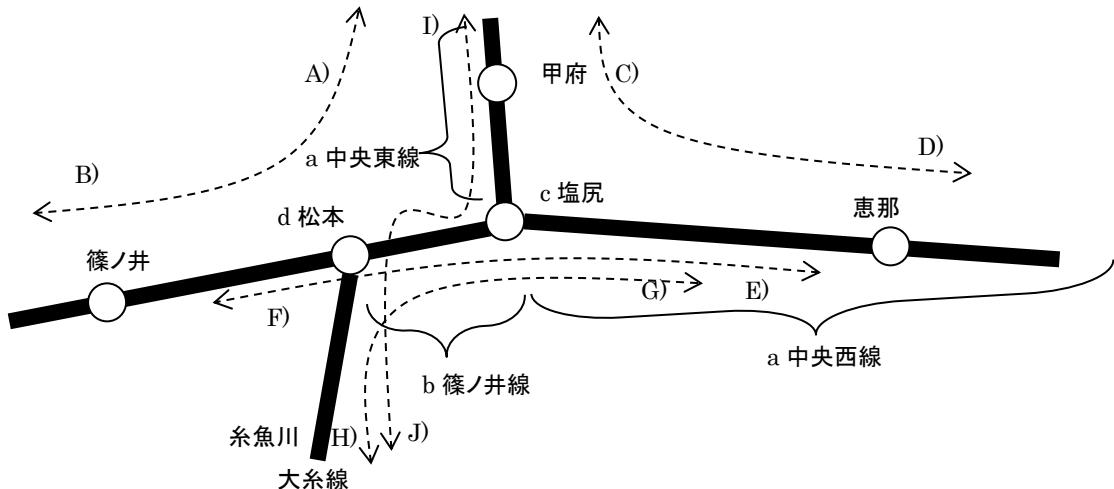
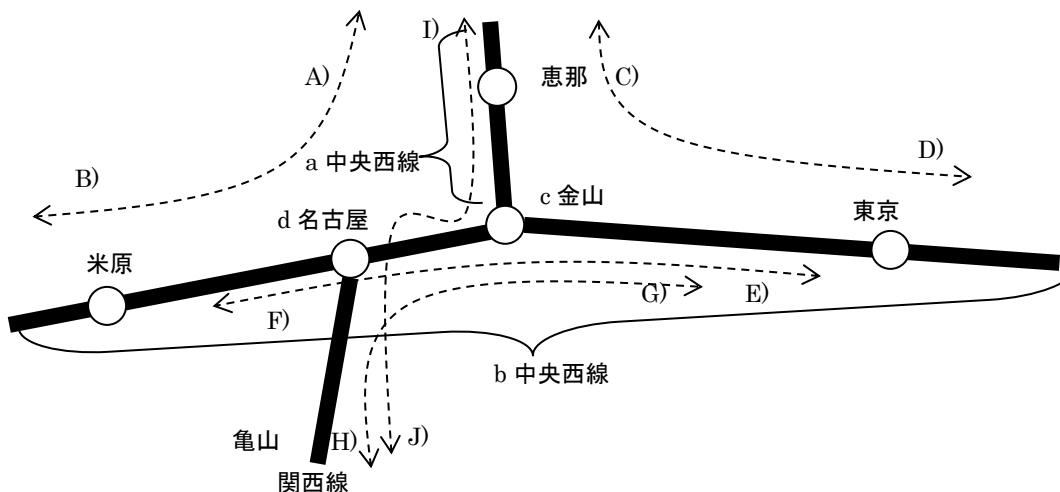
重複路線は配列上存在することを認める。経由表示時に考慮することにする。

RemoveDup()はやらない

\*2 エラーで抜けたときは、フラグリセットする。

また、エラーで抜けた場合、分岐マークフラグも含め適切に戻すことを考慮する。

\*3 lflg の 31 ビット目が ON の場合、removeTail() が呼び出されたとき、その上の 2 回配列を消去する。(→逆に通常、配列に追加時は、lflg の b31 は off して挿入すること)



(b 根室線)			
根室	d 鉄路	c 東鉄路	帶広
			(a 鉄網線)
			網走
(b 宗谷線)			
	d 旭川	c 新旭川	稚内
			(a 石北線)
			上川
(b 函館線)			
小樽	d 札幌	c 白石	江別

		(a 千歳線)	
北広島			
(b 函館線)			
江別	d 札幌	c 桑園	小樽
		(a 札沼線)	
		新十津川	
(b 室蘭線)			
室蘭	d 苫小牧	c 沼ノ端	岩見沢
		(a 千歳線)	
		北広島	
(b 函館線)			
	d 函館	c 五稜郭	長万部
		(a 江差線)	
		木古内	
(b 津軽線)			
青森	d 蟹田	c 中小国	三厩
		(a 海峠線)	
		木古内	
(b 奥羽線)			
秋田	d 弘前	c 川部	青森
		(a 五能線)	
		五所川原	
(b 奥羽線)			
	d 秋田	c 追分	青森
		(a 男鹿線)	
		男鹿	
(b 奥羽線)			
福島	d 山形	c 羽前千歳	秋田
		(a 仙山線)	
		仙台	
(b 奥羽線)			
福島	d 山形	c 北山形	秋田
		(a 左沢線)	
		左沢	
(b 東北線)			
大宮	d 郡山	c 安積永盛	仙台
		(a 水郡線)	
		水戸	
(b 羽越線)			
秋田	d 酒田	c 余目	新津
		(a 陸羽西線)	
		新庄	
(b 信越線①)			
新潟	d 長岡	c 宮内	直江津
		(a 上越線)	
		高崎	
(b 東北線②)			
大宮	d 宇都宮	c 宝積寺	仙台
		(a 烏山線)	
		烏山	
(b 上越線②)			
	d 高崎	c 新前橋	宮内
		(a 両毛線)	
		小山	
(b 高崎線②)			
	d 高崎	c 倉賀野	大宮
		(a 八高線)	
		八王子	

(b 東海道線)			
神戸	d 横浜	c 東神奈川	東京
		(a 横浜線)	
八王子			
(b 東北線*)			
		c 神田	仙台
		(a 中央東線)	
塩尻			
(b 山手線*)			
池袋	d 新宿	c 代々木	大崎
		(a 中央東線)	
御茶ノ水			
(b 篠ノ井線)		中央東線	
		c 塩尻	新宿
		(a 中央西線)	
恵那			
(b 東海道線)			
大阪	d 名古屋	c 金山	横浜
		(a 中央西線)	
恵那			
(b 北陸線)			
富山	d 金沢	c 津幡	敦賀
		(a 七尾線)	
和倉温泉			
(b 北陸線)			
金沢	d 福井	c 越前花堂	米原
		(a 越美北線)	
九頭竜湖			
(b 北陸線)			
米原	d 敦賀	c 近江塩津	富山
		(a 湖西線)	
山科			
(b 東海道線)			
神戸	d 京都	c 山科	横浜
		(a 湖西線)	
近江塩津			
(b 東海道線)			
横浜	d 大阪	c 尼崎	神戸
		(a 福知山線)	
福知山			
(b 山陽線)			
下関	d 岡山	c 東岡山	神戸
		(a 赤穂線)	
播州赤穂			
(b 山陽線)			
神戸	d 岡山	c 倉敷	下関
		(a 伯備線)	
新見			
(b 伯備線)			
倉敷	d 新見	c 備中神代	伯耆大山
		(a 芸備線)	
広島			
(b 伯備線)			
伯耆大山	d 備中神代	c 新見	倉敷
		(a 姫新線)	
姫路			
(b 山陰線)			
益田	d 米子	c 伯耆大山	京都

		(a 伯備線)	
新見			
(b 予讃線)			
松山	d 丸亀	c 宇多津	高松
		(a 瀬戸大橋線)	
		児島	
(b 予讃線)			
高松	d 丸亀	c 多度津	松山
		(a 土讃線)	
		高知	
(b 高徳線)			
佐古	d 勝瑞	c 池谷	高松
		(a 鳴門線)	
		鳴門	
(b 高徳線)			
	d 徳島	c 佐古	高松
		(a 徳島線)	
		佃	
(b 土讃線)			
高知	d 阿波池田	c 佃	多度津
		(a 徳島線)	
		佐古	
(b 予讃線)			
松山	d 伊予市	c 向井原	宇和島
		(a 予讃線：向井原-内子)	
		内子	
(b 予讃線)			
	d 字和島	c 北宇和島	高松
		(a 予土線)	
		若井	
(b 山陽線)			
下関	d 広島	c ※海田市	神戸
		(a 吳線)	
		吳	
(b 山陽線)			
神戸	d 広島	c 横川	下関
		(a 可部線)	
		可部	
(b 山陽線)			
	d 下関	c 幡生	神戸
		(a 山陰線)	
		松江	
(b 鹿児島線)			
門司	d 小倉	c 西小倉	博多
		(a 日豊線)	
		宮崎	
(b 鹿児島線)			
熊本	d 博多	c 吉塚	門司
		(a 篠栗線)	
		篠栗	
(b 長崎線)			
鳥栖	d 佐賀	c 久保田	長崎
		(a 唐津線)	
		唐津	
(b 日豊線)			
門司	d 小倉	c 城野	宮崎
		(a 日田彦山線)	
		夜明	

(b) 久大線)			
大分	d 日田	c 夜明	久留米
		(a) 日田彦山線)	
城野			
(b) 鹿児島線)			
博多	d 熊本	c 宇土	八代
		(a) 三角線)	
三角			
(b) 日南線)			
宮崎	d 南宮崎	c 田吉	志布志
		(a) 宮崎空港線)	
宮崎空港			

## ■ 新幹線と在来線乗り換え

### 新幹線の分岐駅

データベース上では、新幹線は並行在来線とは別線として扱う。ただし、並行在来線内の分岐駅も`t_lines`に含める。乗換路線リストに表示させないように、`sales_km<0`以外をクエリー条件に含めた結果レコードセットのみを表示させる。

```
品川 >
新横浜
    東神奈川
    横浜
    大船
    茅ヶ崎
    国府津
小田原 <
    熱海
三島 >
    沼津
    富士
新富士
静岡 <

京都
    山科+
    草津+
米原 >
岐阜羽島
    大垣
    岐阜
名古屋 <
    金山+
    大府+
三河安城
豊橋
浜松
静岡

> 在来線と別線扱い開始
< 在来線と別線扱い終了
+ 路線テーブルに含める
(インデントされた駅) :新幹線停車駅のない在来線分岐駅
```

### 新幹線と並行在来線の別線扱い区間

新幹線と在来線が並行している区間では、同じ線として営業キロ等の計算をします。

- ・東海道・山陽新幹線＝東海道本線・山陽本線・鹿児島本線
- ・九州新幹線＝鹿児島本線

- ・東北新幹線＝東北本線
  - ・上越新幹線＝東北本線・高崎線・上越線・信越本線
- ただし、次の区間内の各駅(両端の駅を除く)を発駅もしくは着駅または接続駅とする場合は、別の線として営業キロ等の計算をします。

東海道・山陽新幹線と在来線	品川～小田原、三島～静岡、名古屋～米原、新大阪～西明石、福山～三原、三原～広島、広島～徳山
九州新幹線と在来線	博多～久留米、筑後船小屋～熊本
東北新幹線と在来線	福島～仙台、仙台～一ノ関、一ノ関～北上、北上～盛岡
上越新幹線と在来線	熊谷～高崎、高崎～越後湯沢、長岡～新潟

## データ定義と実装

spe_route	
ビット位置	説明
BSRSHINKTRS	新幹線駅で平行在来線乗り換え時の侵入許可フラグ
BSRSHINKTRSLW	00=通常(三河安城、豊橋、浜松、掛川、熱海、新横浜、東京、新富士など) 01=下り線、平行在来線上り有効(静岡など) 10=上り線、平行在来線下り有効(名古屋、三島など) 11=上下線、平行在来線上下有効(広島など)

### Ex.1 下り新幹線(通常駅)→上り在来線

	東京
東海道新幹線	浜松
東海道	天竜川

下り  
上り

新幹線 浜松-掛川は在来線同一視なので後戻りとなり、NG (新幹線下り降車駅浜松の  
spe\_route. BSRSHINKTRSLW=x0)

### Ex.2 上り新幹線(通常駅)→上り在来線

	米原
東海道新幹線	浜松
東海道	天竜川

上り  
上り

上りのみなので、OK(後戻りしていない) (新幹線 米原-浜松は上り。東海道 浜松-天竜川も上り)

参考：新幹線分岐駅：熱海-三島-静岡-掛川-浜松

**Ex.3 下り新幹線(別線端駅)→上り在来線**

	東京
東海道新幹線	静岡
東海道	草薙

下り  
上り

静岡で下り→上り乗車になるが新幹線 静岡の spe\_route. BSRSHINKTRSALW=x1 なので OK

**Ex.4 上り新幹線(別線端駅)→下り在来線**

	米原
東海道新幹線	静岡
東海道	焼津

上り  
下り

静岡で上り→下り乗車になるが新幹線 静岡の spe\_route. BSRSHINKTRSALW=0x なので NG

**Ex.5 上り新幹線(別線端駅)→上り在来線**

	名古屋
東海道新幹線	静岡
東海道	草薙

上り  
上り

上りのみなので、OK

---

**Ex.6 下り在来線→上り新幹線(通常駅)**

	天竜川
東海道	浜松
東海道新幹線	東京

下り  
上り

Ex.7 の草薙-静岡-東京と同一パターンだが、新幹線 浜松の spe\_route. BSRSHINKTRSALW=x0 なので NG

**Ex.7 上り在来線→上り新幹線(通常駅)**

	焼津
東海道	静岡
東海道新幹線	東京

上り  
上り

上りのみなので、OK(後戻りしていない) (東海道 焼津-静岡。新幹線 静岡-東京は上り。も上り)

### Ex.8 下り在来線→上り新幹線(別線端駅)

	草薙
東海道	静岡
東海道新幹線	東京

下り

上り

静岡で下り→上り乗車になるが新幹線 静岡の spe\_route. BSRSHINKTRSLW=x1 なので OK

### Ex.9 上り在来線→下り新幹線(別線端駅)

	焼津
東海道	静岡
東海道新幹線	米原

上り

下り

静岡で下り→上り乗車になるが新幹線 静岡の spe\_route. BSRSHINKTRSLW=0x なので NG

### Ex.10 上り在来線→上り新幹線(別線端駅)

	焼津
東海道	静岡
東海道新幹線	東京

上り

上り

上りのみなので、OK

	神戸
東海道	新大阪
東海道新幹線	東京

上り

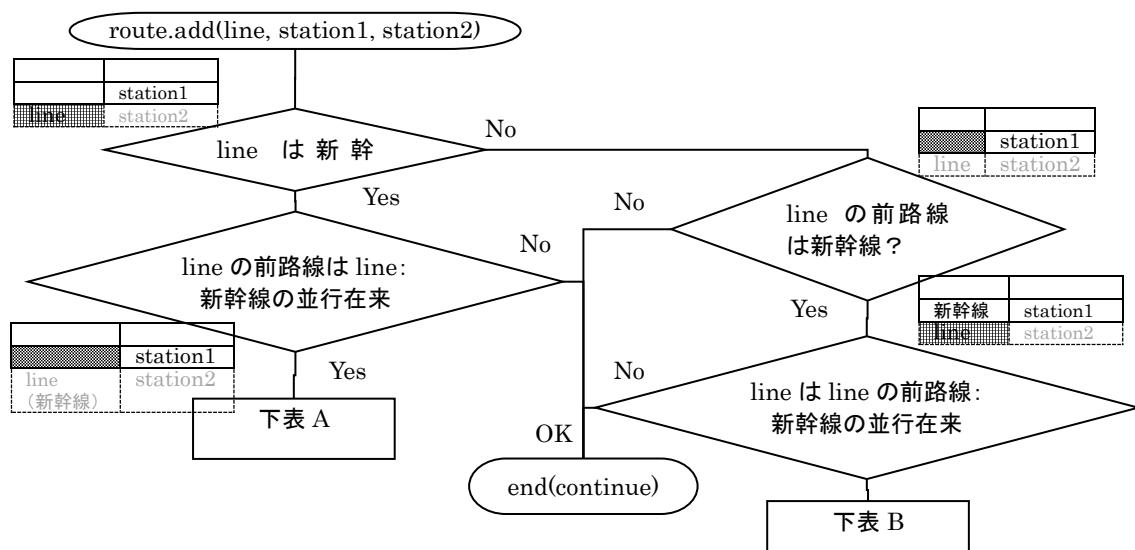
上り

	博多
鹿児島	小倉
山陽新幹線	東京

上り

上り

Route.add(line, station1, station2)



(表 A)

line は新幹線で、前路線(routeItem[-1])は新幹線の並行在来線の場合(Ex.6~Ex.10)

line の 前 路 線 (処理)

station1 → station2	(routeItem[-1])の 発 駅 (routeItem[-2])→ station1	
下り	下り	OK
上り	上り	
下り	上り	前路線着駅(routeItem[-1].station)=station1 の spe_route.BSRSHINKTRSALW=1x の場合?OK : NG
上り	下り	前路線着駅(routeItem[-1].station)=station1 の spe_route.BSRSHINKTRSALW=x1 の場合?OK : NG

a) OK

	岐阜
東海道	名古屋
東海道新幹線	米原

上り  
下り

b) OK Ex.8

	草薙
東海道	静岡
東海道新幹線	東京

下り  
上りa)  
b)

(表 B)

前路線(routeItem[-1])は新幹線で line は前路線の並行在来線の場合(Ex.1~Ex.5)		
前 路 線 (routeItem[-1])の 発 駅 (routeItem[-2])→ station1	line の station1→ station2	(処理)
下り	下り	OK
上り	上り	
下り	上り	並行在来線の着駅: station2 の spe_route. BSRSHINKTRSALW=1x の場合?OK : NG
上り	下り	並行在来線の着駅: station2 の spe_route. BSRSHINKTRSALW=x1 の場合?OK : NG

c) OK

	米原
東海道新幹線	名古屋
東海道	岐阜

上り  
下り

d) OK Ex.3

	東京
東海道新幹線	静岡
東海道	草薙

下り  
上りc)  
d)

OK : 次のチェックへ

NG : route.add() 処理を失敗(-1)で返す。routeItem に追加しない。

## ● 新幹線の平行在来線を得る

```
int getHeikouZairai(line, station1, station2)
    line      新幹線
    station1 並行在来線駅(新幹線接続駅)
    station2 方面
```

戻り値: =0 並行在来線ではない  
≠0 並行在来線 line\_id  
MSB=1: 上り／0: 下り ←いる?

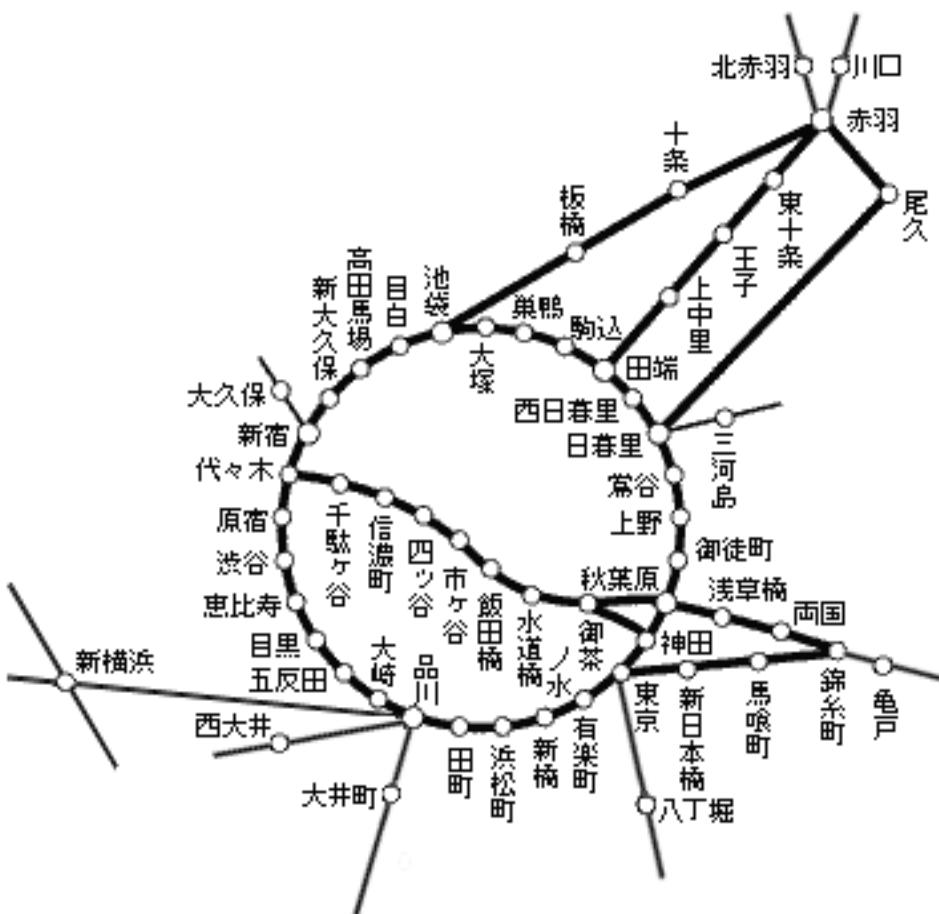
```
select_spe_route and 0x1ff
from   t_lines
where  line_id = :line      -- 新幹線
and    station_id=:station  -- の接続駅は spe_route.b8-0 に在来線 line_id
```

```
select_spe_route -の bit BSRSHINKTRSLW を取り出す
from   t_lines
where  line_id = :line      -- 在来線
and    station_id=:station  -- 新幹線の接続駅
and    spe_route.BSRSHINKTRS=1
```

## ■ 規則 70 条 最短経路で計算

下図にしめされる駅を通過する場合、実際の乗車路線に関わらず、最短経路で計算する（重複・復乗禁止）。

発着が近郊区間内駅(新幹線未乗車)相互発着の場合、そちらに準拠する



#### ○ 70条適用駅(図の太線内駅)

70 条の適用は、69 条の適用後に実施すること

70条通過のみで、発または着のいずれかは適用されない。100km/200kmを越える場合は、87, 86条が適用される。

70条通過が複数あった場合、適用しない。

159、160 条は乗車券の効力の問題なので指定経路通りで計算する（自動経路ボタンのサジェスチョンはしている）

\* 70条駅が2駅のみの場合も「田端 品川(山手線)」の例もあるので、これも変換する。

~~\*\* 京葉線で東京を経て総武線で千葉へ行く経路や、仙台・日暮里・水戸などは良いが、名古屋(東海道線)東京(中央線)甲府などの場合、品川・新宿としなければならない。~~

~~関数 stationId -> rule70max(int lineId, int stationId1, int stationId2)~~  
~~路線、駅1、駅2の間の70条適用端駅を得る。駅1は70条適用駅で、駅2は70条非適用駅である必要があります。~~

品川、新宿、赤羽、日暮里、錦糸町、東京(優先順位の並び)の各駅の相互組み合わせのみ定義する。  
(15通り)

## 実装概要

大環状線内エリアの進入・脱出路線から、脱出駅、进入駅と、その区間接続最短距離に置き換える。

## データ定義

テーブル t_station.city_flg	
ビット	説明
BCRULE70	1ビット: 1で70条適用駅

テーブル t_line.		
フィールド	ビット	説明
spe_route	BSR70	1ビット: 1で70条定義レコードであることをしめします

### テーブル定義（一部）

station_id1	station_id2	sales_km
品川	新宿	0
品川	赤羽	42
品川	日暮里	0
品川	錦糸町	58
品川	東京	0
新宿	赤羽	0
新宿	日暮里	0
新宿	錦糸町	113
:	:	:

70条定義の外側の駅:品川、新宿、赤羽、日暮里、錦糸町、東京の6駅の  $nCr = (6*5)/2 = 15$  通りのテーブルを用意する。

## クエリー

関数 int RetrieveRule70StationOfOutside(int lineId, int stationId1, int stationId2)

路線内の stationId1 から stationId2 間の最も外側の 70 条適用駅を得る。

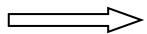
必ず、stationId1 は、70 条適用駅であり、stationId2 は 70 条非適用駅である必要がある。

また、stationId1, stationId2 は lineId 路線内に所属する駅でなくてはならない。

## 実装

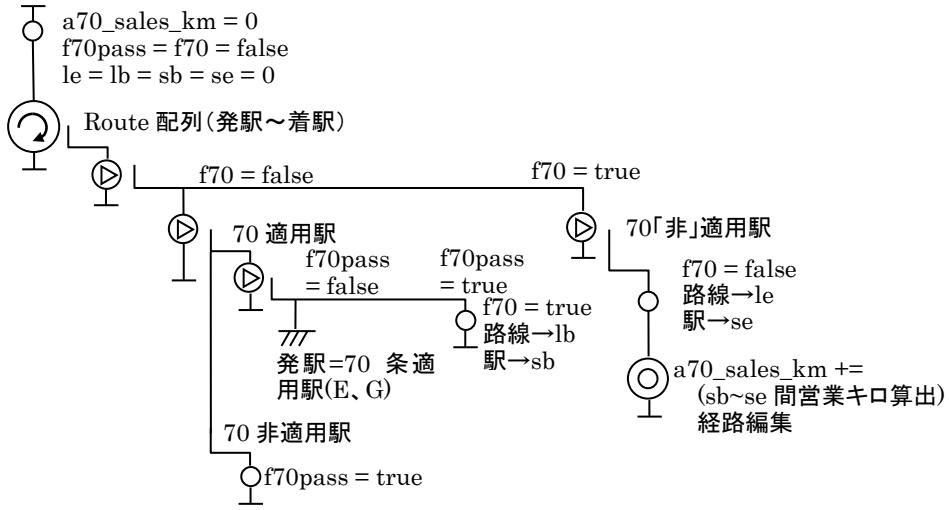
運賃計算時に判定する。

※ 本 70 条適用後、69 条の適用後、さらに本 70 条適用(常磐-京葉など)

	適用パターン	
A		対象外(86、87 条適用)
B		対象外
C		適用
D		対象外(86、87 条適用)
E		対象外(86、87 条適用)
F		対象外(○内最短経路)
G		対象外(86、87 条も非適用)
H		対象外(86、87 条も非適用)

### ● 営業規則 70 条のチェック

```
int Route::ReRouteRule70j(const vector<RouteItem>& in_route_list,  
vector<RouteItem>* out_route_list)
```



東海道-中央

乗車経路

	川崎
東海道	東京
中央東	八王子

データ構造

	川崎
東海道	品川
1 rule 70	新宿
中央	八王子

## ＜処理フロー＞

- 1) t\_lines テーブルより、spe\_route.BSR70 ON(70条適用駅)の外側駅を得る。  
「東海道線」川崎～東京間でクエリーを実行し「品川」を取得。「中央東線」東京～八王子間でクエリーを実行し「新宿」を得る。
  - 2) Route 配列の入口駅を品川に変更し、出口路線、駅の前に、出口路線「70条適用駅の外側駅」即ち、l\_rule70:新宿を挿入する。  
l\_rule70 は、70条の特別路線で値 (line\_id) を、-10 とする。  
経由表示は、line\_id=-10 で路線は得られない為、表示をせず、  
経由(東海道、品川、新宿、中央)となる。

中央・東海道

乗車経路

	八王子
中央東	東京
東海道	川崎

70 条適用後データ構造

	<b>八王子</b>
<b>中央東</b>	<b>新宿</b>
<b>1 rule 70</b>	<b>品川</b>
<b>東海道</b>	<b>川崎</b>

「中央東」八王子～東京間:新宿→中央東の着駅を東京→新宿に置換

「東海道」東京～川崎：品川→東海道の前に<sub>1</sub> rule70-品川を挿入する。

### 乗車経路

	八王子
中央東	御茶ノ水
総武 2	錦糸町
総武	東京
東海道	川崎

### 70条適用後データ構造

	八王子
中央東	新宿
l_rule70	品川
東海道	川崎

「中央東」八王子～御茶ノ水間：新宿→中央東の着駅を御茶ノ水→新宿に置換

「中央東」～「東海道」の間の経由は全削除(進入駅～脱出駅の中間路線の削除)

「東海道」東京～川崎：品川→東海道の前に l\_rule70-品川を挿入する。

### 東北-京葉(蘇我以遠)

#### 乗車経路

	宇都宮
東北	田端
山手	代々木
中央東	東京
京葉	蘇我

#### 70条→69条適用後経路

	宇都宮
東北	田端
山手	代々木
中央東	東京
総武線	千葉
外房線	蘇我

#### 70条適用後データ構造

	宇都宮
東北	赤羽
l_rule70	錦糸町
総武線	千葉
外房線	蘇我

※ 69条適用は70条適用の前におこなう必要がある。

#### 69条適用

「東北」宇都宮～東京間：赤羽、→東北の着駅を田端→赤羽に置換

「東北」～「総武線」の間の経由は全削除(進入駅～脱出駅の中間路線の削除)

「総武線」東京～千葉：錦糸町→総武線の前に l\_rule70-錦糸町を挿入する。

### 乗車経路

	静岡
東海道	東京
京葉	蘇我

#### 70条→69条適用後経路

	静岡
東海道	東京
総武線	千葉
外房線	蘇我

#### 70条適用後データ構造

	静岡
東海道	品川
l_rule70	錦糸町
総武線	千葉
外房線	蘇我

69変換後、進入区間である、東海道線：静岡-東京の外側 70条適用駅(品川)に置換

脱出区間は、総武線：東京-千葉の外側 70条適用駅(錦糸町)を脱出区間の前に挿入する

进入区間と脱出区間の間の区間はないため削除はない

### 乗車経路

	静岡
東海道	東京
京葉	新木場

#### 70条適用後データ構造

	静岡
東海道	品川
l_rule70	東京
京葉	新木場

### 常磐-京葉(蘇我以前)

#### 乗車経路

	いわき
常磐	日暮里

#### 70条適用後データ構造

	いわき
常磐	日暮里

東北	東京
京葉	葛西

1 rule70	東京
京葉	葛西

### 常磐-京葉(蘇我以遠)

乗車経路

	いわき
常磐	日暮里
東北	東京
京葉	蘇我

70条→69条適用後経路

	いわき
常磐	日暮里
東北	東京
総武線	千葉
外房線	蘇我

70条適用後データ構造

	いわき
常磐	日暮里
1 rule70	錦糸町
総武線	千葉
外房線	蘇我

### 常磐-上越新幹線 1

乗車経路

	いわき
常磐	日暮里
東北	大宮
上越新幹線	高崎

70条適用後データ構造

	いわき
常磐	日暮里
1 rule70	赤羽
東北	大宮
上越新幹線	高崎

※ 日暮里-東京-上越新幹線は重複経路で指定不可

### 常磐-上越新幹線 2

乗車経路

	いわき
常磐	日暮里
東北	品川
山手	池袋
赤羽	赤羽
埼京	大宮
上越新幹線	高崎

70条適用後データ構造

	いわき
常磐	日暮里
1 rule70	赤羽
埼京	大宮

上越新幹線 高崎

	高崎			高崎
上越新幹線	大宮		上越新幹線	赤羽
東北線	日暮里		1 rule70	日暮里
常磐線	いわき		常磐線	いわき

※ 経由に表示する駅としない駅をフラグで制御する。赤羽などは表示に載せない。

### 京葉(蘇我以近)-中央

乗車経路

	葛西
京葉	東京
中央東	八王子

70条適用後データ構造

	葛西
京葉	東京
1 rule70	新宿
中央東	八王子

### 京葉(蘇我以遠)-中央

乗車経路

	蘇我
京葉	東京
中央東	八王子

70条→69条→157条適用  
後経路

	蘇我
外房	千葉
総武	東京
中央東	八王子

70条適用後データ構造

	蘇我
外房	千葉
総武	錦糸町
l_rule70	新宿
中央東	八王子

### 総武(蘇我以遠)-京葉

乗車経路

	大網
外房	千葉
総武	錦糸町
総武(錦糸町-御茶ノ水)	秋葉原
東北	東京
京葉	蘇我

70条適用後データ構造

	大網	
外房	千葉	
総武	錦糸町	
l_rule70	東京	
京葉	蘇我	

↑このような6の字切符が買えるか！！テストパターンに含めよ！！

### 常磐-東北(埼京線)1

	いわき			いわき
常磐	日暮里		常磐	日暮里
東北	品川		l_rule70	赤羽
山手	池袋		埼京	武藏浦和
赤羽	赤羽		武蔵野	西国分寺
埼京	武藏浦和			
武蔵野	西国分寺			

### 常磐-東北(埼京線)2

	いわき
常磐	日暮里
東北	品川
山手	池袋
赤羽	赤羽
埼京	武藏浦和
武蔵野	新松戸

常磐	いわき
l_rule70	日暮里
埼京	赤羽
武蔵野	武藏浦和
	新松戸

	仙台
東北線	東京
東海道線	名古屋

	仙台
東北線	赤羽
l_rule70	品川
東海道線	名古屋

	仙台
東北線	東京

	仙台
東北線	東京

※ 70条適用駅で止まっているので(通過していない)ありえない(対象外)

	仙台
東北線	赤羽
埼京線	池袋
山手線	品川
東海道線	名古屋

	仙台
東北線	赤羽
1 rule70	品川
東海道線	名古屋

	大宮
埼京線	赤羽
東北線	南浦和

	大宮
埼京線	赤羽
1 rule70	赤羽
東北線	南浦和

70 条適用駅が一回通過しているだけ

→ 挿入結果として同一駅(赤羽)なら挿入しない

	葛西
京葉線	東京
総武線	西船橋

	葛西
京葉線	東京
1 rule70	錦糸町
総武線	西船橋

↑このような6の字切符が買えるか！！テストパターンに含めよ！！

	葛西
京葉線	東京
東北線	秋葉原
総武線 2	西船橋

	葛西
京葉線	東京
1 rule70	錦糸町
総武線 2	西船橋

	葛西
京葉線	東京
中央東	御茶ノ水
総武線 2	秋葉原
総武線	西船橋

	葛西
京葉線	東京
1 rule70	錦糸町
総武線	西船橋

	新松戸
常磐線	日暮里
東北線	東京
京葉線	蘇我

	新松戸
常磐線	日暮里
1 rule70	東京
京葉線	蘇我

	亀戸
総武線	錦糸町
総武線 2	御茶ノ水
中央東	東京
京葉	蘇我

	亀戸
総武線	錦糸町
1 rule70	東京
京葉	蘇我

	亀戸
総武線	錦糸町
総武線 2	御茶ノ水
中央東	東京
総武線	馬喰町

※ 馬喰町は 70 条適用駅であり、70 条適用駅で終わっている場合は、非適応

	仙台
東北線	東京
中央東線	甲府
身延線	国母

	仙台
東北線	赤羽
l_rule70	新宿
中央東線	甲府
身延線	国母

RetrieveOut70station(lineId, stationId1, stationId2)

一番左側の 70 条適用駅を得る

-stationId1 は 70 条適用駅である必要がある

-stationId2 は 70 条非適用駅である必要がある

→不要。パラメータは路線のみで良し

(大阪市内などが導入されたら、引数に市内 ID を追加することになるか、別関数になるか。多分、別関数になるのか)

クエリー

```
select station_id from t_lines where line_id=?1 and
sales_km=(select max(sales_km) from t_lines where line_id=?1 and
(lflg&(1<<6))!=0);
```

70 条は東京都区内のみが対応なので sales\_km は大きくなる程、外側の 70 条適用駅とすることができるため、クエリーは上記のようになる。

将来、大阪市内の通過のルールができた場合、考慮しなおし(フラグ、DB 構造、C ロジックも含め)

New ReRouteRule70j()

```
stage = 0
route_item_1 = route_item;
stationId = null
stationId_o70 = null
Route Item の列挙(iterator:route_item)
    RouteItem ri <- in_route_list.item
    skip = false
    if stage == 0
        if 70 条適用駅?
            stage = -1           /* 70 条適用エリア内発は非適用 (86、87
条) */
        else
            stage = 1           /* 発駅は 70 条非適用駅 */
            stationId = ri.stationId
    else if stage == 1
        if 70 条適用?
            stage = 2           /* 70 条非適用駅->70 条適用駅に進入
*/
            stationId_o70 = RetrieveOut70station(ri->lineId,
ri->stationId, stationId)
            ASSERT(0 < stationId_o70)
            ri.stationId = stationId_o70
        else
```

```

        stationId = ri.stationId
        // 外のまま
    else if stage == 2
        if 70 条適用? /* 70 条非適用駅->70 条適用駅内 */
            skip = true
            stationId = ri.stationId
            // 中のまま
        else /* 70 条非適用駅->70 条適用駅->70 条
適用駅(70 条適用区間通過) */
            stationId_tmp = RetrieveOut70station(ri->lineId,
stationId, ri->stationId)
            ASSERT(0 < stationId_tmp)
            if stationId_o70 != stationId_tmp
                out_route_list
addtail:RouteItem(l_rule70, stationId_tmp)
            else
                skip = true
            stage = 3
        else if stage == 3
            if 70 条適用?
                stage = 4 // 進入して脱出して(通過し
て)また進入したら 70 条非適用
                out_route_list clear
                out_route_list copy:in_route_list
                return 0
            else
                // 70 条通過(通過して外に出ている)
                // do nothing
        else if stage == -1
            // do nothing

        if !skip
            out_route_list.item <- ri

// 非適合
// 0: 70 条適用駅発
// 1: 70 条適用駅なし(都内を通らない多くのルート)
// 2: 70 条適用駅着(発駅は非適用駅)
// 4: 70 条適用駅区間を通過して再度進入した場合
    in_route_list <- out_route_list
// 適合
// 3: 70 条適用駅通過

```

進入駅なら、発駅～着駅(進入駅)の外側の駅を取得し、駅を置き換える。置き換え前の駅 (a) も保持  
(b) → 置き換え前の駅も置き換え後の駅も保持(置き換え後の駅は 2 つ保存し、一つは更新・上書きされる)

(b) のイテレータでないもので 70 条適用駅は削除(その適用駅を (a) の更新として保持))

70 条非適用駅になったら、発駅(70 条適用駅)(置き換え前の (a) の駅)～着駅(70 条非適用駅)の外側の駅を取得し、

「l\_rule70:70 条外側駅」を挿入

(この外側駅が (a) と同様であったならば、行の挿入自体をおこなわない)  
->埼京、赤羽、東北の V 字型

## ■ 規則 69 条最短経路で計算

### 規則 69 条

	乗車経路			計算経路		
	発	路線	至	計算路線	発	至
1	大沼	函館線(東森経由)	森	函館線	大沼	森
2	日暮里	東北線(尾久経由)	赤羽	東北線	日暮里	赤羽
3	赤羽	東北線(埼京線)	大宮	東北線	赤羽	大宮
4	品川	東海道線(西大井経由)	鶴見	東海道線	品川	鶴見
5	東京	京葉線	蘇我	総武線 外房線	東京	千葉 蘇我
6	山科	東海道線	米原	湖西線	山科	近江塩津
	米原	北陸線	近江塩津			
*	大阪	大阪環状線(天満経由)	天王寺	大阪環状線(福島経由)	大阪	天王寺
7	三原	吳線	海田市	山陽線	三原	海田市
8	岩国	山陽線	櫛ヶ浜	岩徳線	岩国	櫛ヶ浜
9	秋葉原	東北線	東京	総武線(錦糸町-御茶ノ水)	秋葉原	錦糸町
	東京	総武線	錦糸町			
10	御茶ノ水	中央東線	神田	総武線(錦糸町-御茶ノ水)	御茶ノ水	錦糸町
	神田	東北線	秋葉原			
10	御茶ノ水	中央東線	東京	総武線(錦糸町-御茶ノ水)	御茶ノ水	錦糸町
	東京	総武線	錦糸町			

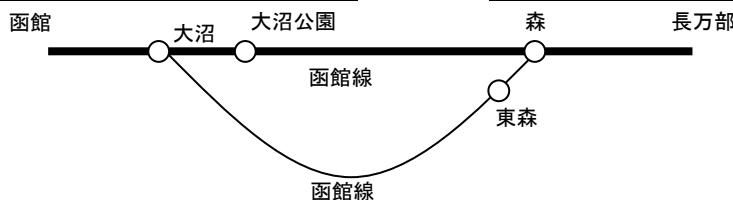
\* 大阪環状線は大阪・天王寺は計算時必然的に最短で計算されるので特別処理は不要

※ 「御茶ノ水」(中央東線)「東京」(東北線)「秋葉原」は、東京・秋葉原間に神田が存在するため指定できない

### 1. 大沼・森

	函館
函館線	大沼
函館線(東森経由)	森
函館線	長万部

	函館
↓	↓
↓	↓
函館線	長万部



検索: 函館線(東森経由)、大沼、森

	函館
函館線	長万部

そのまま

### 2. 日暮里・赤羽

	盛岡
東北線(日暮里経由)	赤羽
東北線	日暮里
東北線	東京

	盛岡
↓	↓
↓	↓
東北線	東京

検索: 東北線(日暮里経由)、赤羽、日暮里

### 3. 赤羽・大宮

	東京
東北線	赤羽
東北線(埼京線)	大宮
東北線	盛岡

	東京
↓	↓
↓	↓
東北線	盛岡

検索: 東北線(埼京線)、赤羽、武藏浦和、大宮

### 4. 品川・鶴見

	東京
東海道線	品川
東海道線(西大井経由)	鶴見
東海道線	横浜

	東京
↓	↓
↓	↓
東海道線	横浜

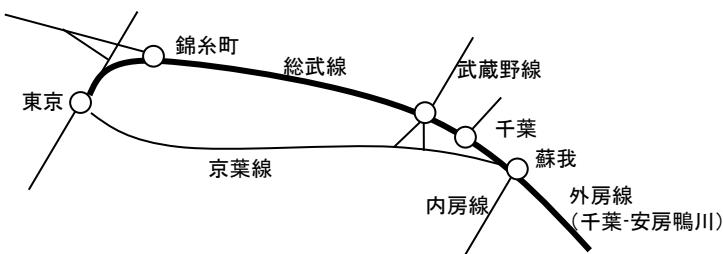
検索: 東海道線(西大井経由)、品川、武藏小杉、鶴見

### 5. 東京・蘇我

	東京
京葉線	蘇我
外房線	勝浦

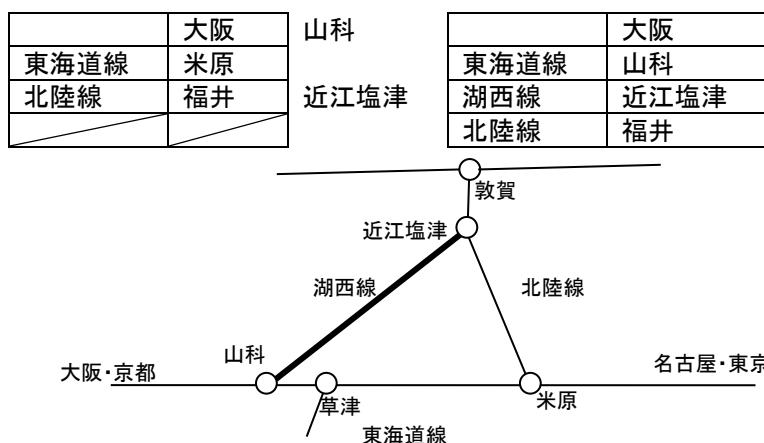
	東京
↓	↓
↓	↓
東海道線	横浜

蘇我



検索：京葉線、東京、市川塩浜、南船橋、蘇我

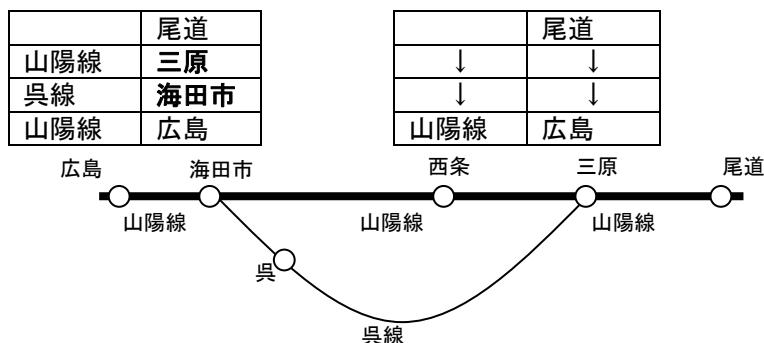
#### 6. 山科・近江塩津



検索：東海道線、山科、草津、米原

北陸線、米原、近江塩津

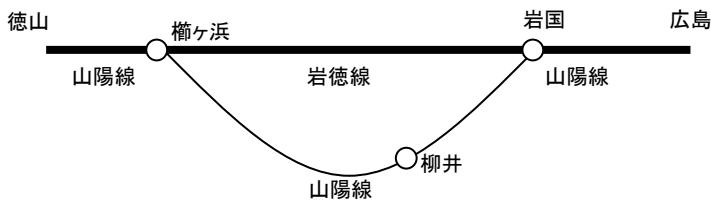
#### 7. 三原・海田市



検索：呉線、海田市、三原

#### 8. 岩国・櫛ヶ浜





検索：山陽線、櫛ヶ浜、岩国

## 実装概要

実装法：

規則 70 条、86、87 条のあとに行う

運賃計算時、経路検索して、置き換える。

このとき、運賃計算は毎回行っているので、`MAX_A69ROUTELINES < 経路配列数のときは、経路配列数 - MAX_A69ROUTELINES` から検査をする(前半のルートは置換済みであるので検索から除外)  
`MAX_A69ROUTELINES=3`

以下の注釈を「注意事項に含める」

「岩国-櫛ヶ浜間は、岩徳線経由でも山陽線(新幹線含む)経由で乗車可能」

## データ定義

### データ定義条件

- `spe_route.BSR69TERM = 1` は、営業規則 69 条の端駅に定義(上記表の「発」、「至」)
- 1 路線に `spe_route.BSR69TERM=1` かつ同一の `spe_route.BSR69NOMASK` が 2 つあると置換する
- 1 路線に `spe_route.BSR69TERM=1` が 1 つもない場合は、営業規則 69 条に該当しないのにもしない。
- 1 路線に `spe_route.BSR69TERM=1` が 1 つしかない場合は、「発」～「至」駅の途中駅から経路を外れているので営業規則 69 条には該当せすなにもしない。
- 1 路線に `spe_route.BSR69TERM=1` が 3 つ以上存在することはない(現状はないが将来あり得ないこともないが、`spe_route.BSR69NOMASK` までは同一ではない)
- 1 路線に `spe_route.BSR69TERM=1` で異なる `spe_route.BSR69NOMASK` を挟むことはない(必ず `spe_route.BSR69TERM=1+spe_route.BSR69NOMASK` は連続する)。
- `spe_route.BSR69TERM=1` は複数路線をまたぐ場合があり得る。この場合、乗り換えて継続する駅を `spe_route.BSR69CONT=1` でしめす(上表では「米原」にのみ該当)。

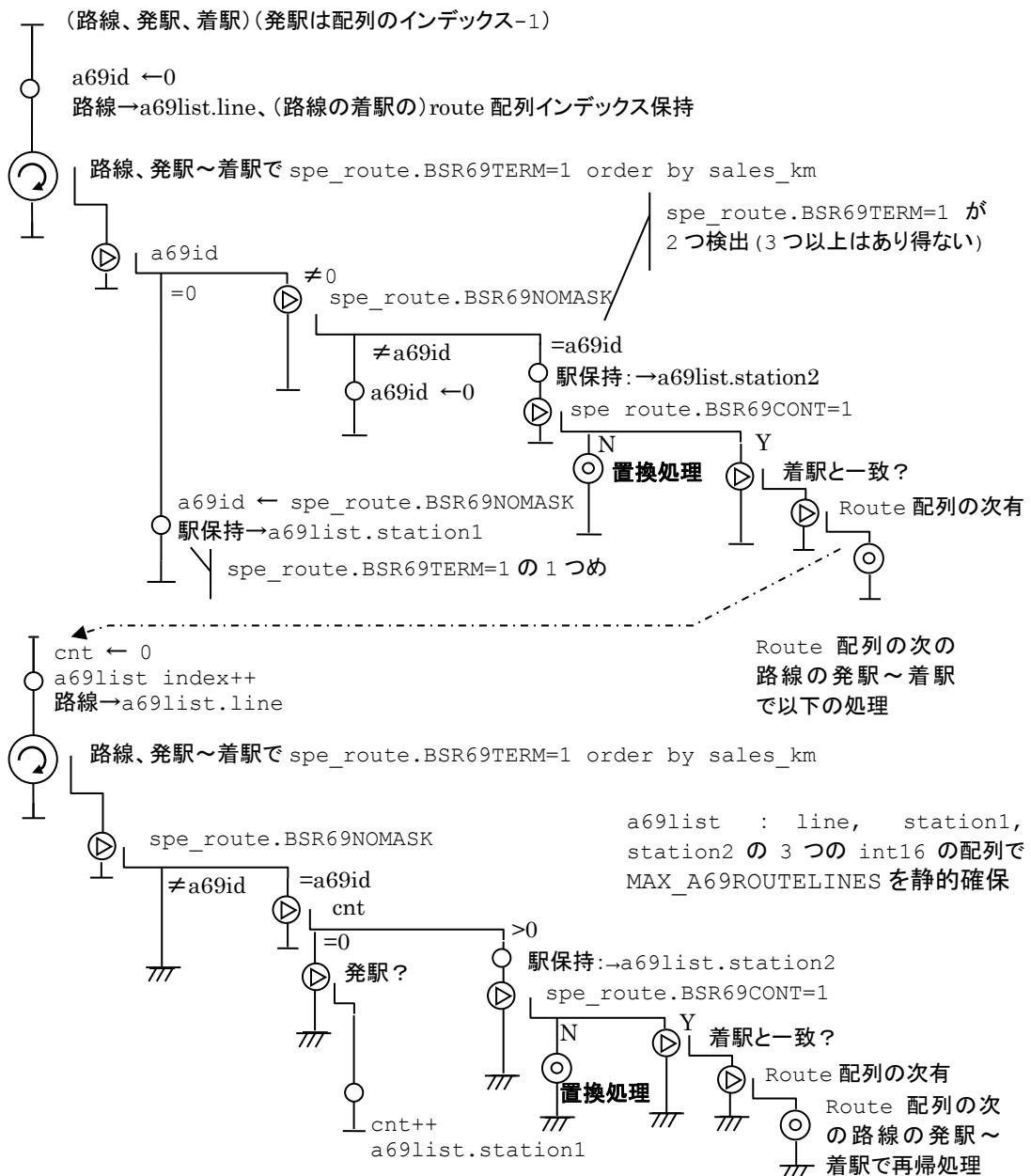
- 一周ルートは適用させないように(総武・京葉ルート大網・蘇我・東京・蘇我や、尾道～三原～呉線～海田市～山陽線～西条への切符)
- さらに総武線のときは、秋葉原経由、東京経由双方とも考慮されている必要がある

テーブル定義 : t\_rule69

id	station_id1	station_id2	line_id	ord
1	大沼	森	函館線	0
2	日暮里	赤羽	東北線	0
3	赤羽	大宮	東北線	0
4	品川	鶴見	東海道線	0
5	東京	千葉	総武線	0
5	千葉	蘇我	外房線	1
6	山科	近江塙津	湖西線	0
7	三原	海田市	山陽線	0
8	岩国	櫛ヶ浜	岩徳線	0

## 実装詳細

## 検索処理



※ a69list 配列は、上表の「乗車経路」の「路線」「発」「至」を保持されることになります。

a69list 配列は、3 つのメンバすべて ≠ 0 で有効であり、1 つでも = 0 があるレコードは無効で終端を表します

### 上記フローを一部クエリー化

HCP 前段をクエリー化>引数:line\_id, station\_id1, station\_id2(路線の駅1～駅2間に BSR69TERM!=0 の駅を抽出)>出力:a69list.station, a69list.lflg(該当駅とその属性を返す)((lflg>>17) & 0x0f) レコードが連続して 2 行返されること)

連続して 2 行に満たない場合、途中下車、乗り換えが考えられるのでチェック対象外

連続して 2 行存在していた場合、置換処理へ、

さらにそのまえに `a69list.station` の 2 レコード目の `((lflg&(1<<BSR69TERM)!=0` の場合 (米原駅を指す)、`station_id2` が一致しているか判定。不一致の場合、上りの場合、東海道線を名古屋方面へ行ってしまっている(下りは湖西線の終点が米原であるためありえない)ので、置換適用外。一致していた場合、`RouteList` の次のレコードの `lineId` の `stationId` からつぎの `stationId` までの間に `BSR69TERM!=0`かつ `((lflg>>17)&15)!=0` の駅がある場合、  
(→:クエリー結果 `a69list`)

#### 呉線型

[呉線]、三原—海田市

→呉線、三原、海田市

#### 岩徳線型

[山陽線]、新山口(—櫛ヶ浜—岩国)—広島

→山陽線、櫛ヶ浜、岩国

#### 米原型

[東海道線]、京都(—山科—)米原—[北陸線]—(近江塩津)—福井

→東海道線、山科、米原

北陸線、米原、近江塩津

#### 京葉線型

[京葉線]、東京—蘇我—[外房線]—勝浦

→京葉線、東京、蘇我

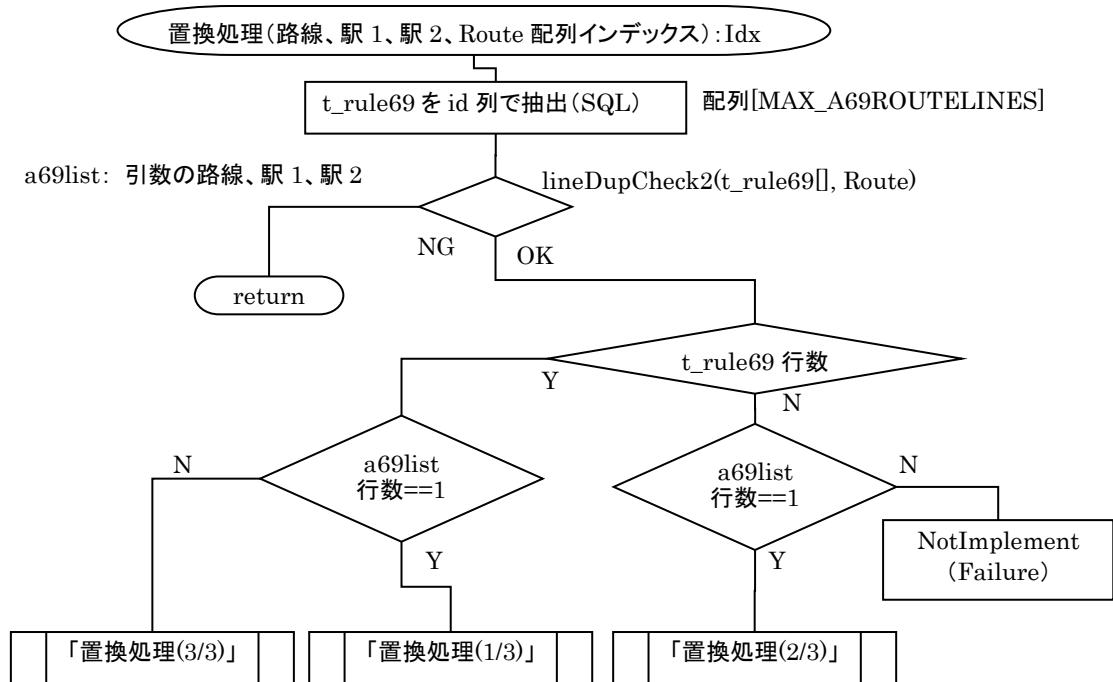
### 置換処理

置換処理は 3 パターン存在する。以下にパターンについてしめす。

	乗車経路	計算経路	例
置換処理 1	1	1	岩国—櫛ヶ浜(山陽線、岩徳線)、三原-海田市(呉線、山陽線)など 多数
置換処理 2	1	N	東京-蘇我(京葉線、総武線・外房線)
置換処理 3	N	1	山科-近江塩津(東海道線・北陸線、湖西線)
—	N	N	例無し(実装しない)※
置換処理未実施	—	—	計算経路(または乗車経路)を経て乗車経路(または計算経路)に乗車する例では置換はおこなわない。 例: 外房線・大網から千葉、総武線を経て東京から京葉線で蘇我へ至るルートの場合、京葉線・東京—蘇我間の置換はおこなわない (但し、上記例の片道乗車券より大網—東京、東京—蘇我という乗車券を分割購入した方が¥40-安い)

※ 現在 N は 2 までしかないので実装は 2 までの対応とする。また、国鉄・JR グループの過去 50 年を振り返っても今後も縮小傾向にあって将来増えることは考えにくい。

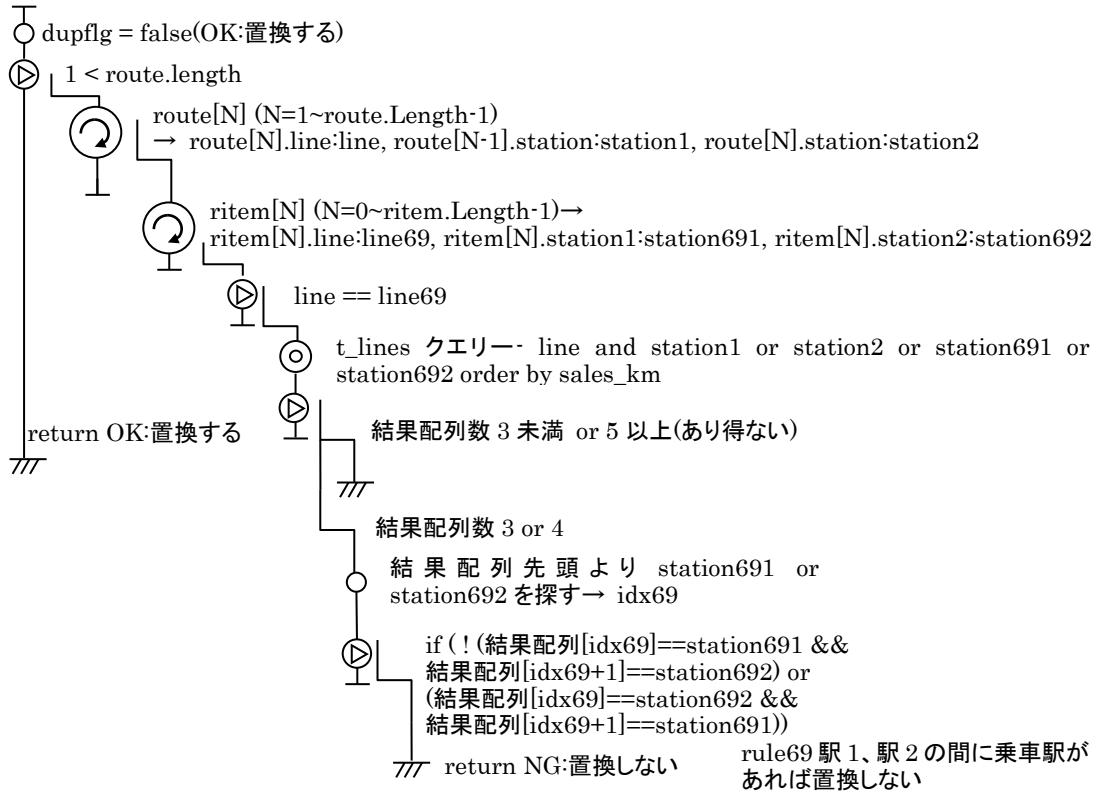
下図は、上記 3 パターンの判定処理。パターン毎の処理詳細については以降にしめします。



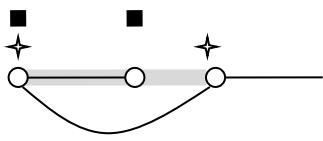
### lineDupCheck2() – 69 条置換判定

(総武-京葉ルート大網-蘇我-東京-蘇我や、尾道～三原～呉線～海田市～山陽線～西条への切符)

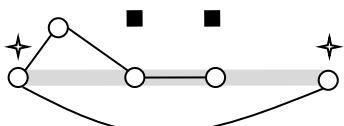
lineDupCheck2(t\_rule69 [] ritem, Route route)



置換しない



置換しない



置換する



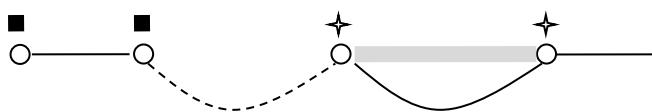
置換経路

他路線

■ 乗車(経路)路線(Route 配列内)

☆ t\_rule69 駅 1 、駅 2(station691, station692)

置換する



(上記フローは破棄)

全 Route で、`route[N].line == t_rule69.line` があった場合 `t_rule69.station1 ~ t_rule69.station2` 内に、`route[N-1].station` と、`route[N].station` のいずれかがあつた場合(`inStation()`)、(但しいずれか一方の場合、その一方が `t_rule69` の端駅と一致の場合置換する)置換適用外とする。



~~全Routeで、route[N].line == t\_rule69.line があった場合 t\_rule69.station1 == t\_rule69.station2 内に、route[N-1].station と route[N].station の 2 駅ともあつた場合(inStation())、置換適用外とする(置換しない)。~~

→ 錦糸町-御茶ノ水-東京-…の例があるので適用していた。

### 置換例 1 三原→海田市(下り)

a69list

吳線	三原	海田市
----	----	-----

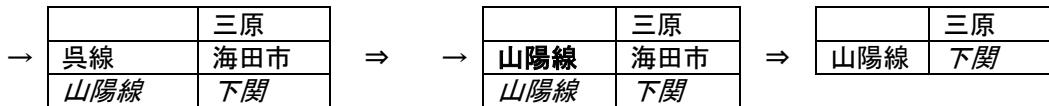
t\_rule69 recordset SQL

山陽線	三原	海田市
-----	----	-----

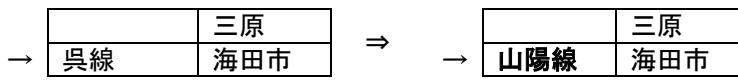
a) 外→外



b) 内→外



c) 内→内



d) 外→内



→ インデックスの指す路線のみを置きかえ

### 置換例 2 下関→神戸(櫛ヶ浜、岩国)(上り)

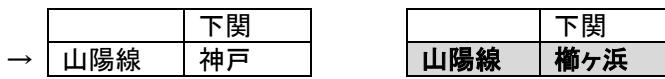
a69list

山陽線	櫛ヶ浜	岩国
-----	-----	----

t\_rule69 recordset SQL

岩徳線	岩国	櫛ヶ浜
-----	----	-----

a) 外→外上り



<b>岩徳線</b>	<b>岩国</b>
山陽線	神戸

b) 内→外

	櫛ヶ浜
→ 山陽線	神戸

	櫛ヶ浜
→ 岩徳線	<b>岩国</b>
山陽線	神戸

(適用外)

	下関
山陽線	櫛ヶ浜

	下関
山陽線	櫛ヶ浜

適用外

c) 内→内

	櫛ヶ浜
→ 山陽線	岩国

	櫛ヶ浜
→ 岩徳線	<b>岩国</b>

d) 外→内

	下関
→ 山陽線	岩国

	下関
→ 岩徳線	<b>岩国</b>

a) 外→外下り

a69list

山陽線	岩国	櫛ヶ浜
t_rule69 recordset SQL		
岩徳線	岩国	櫛ヶ浜

	神戸
→ 山陽線	下関

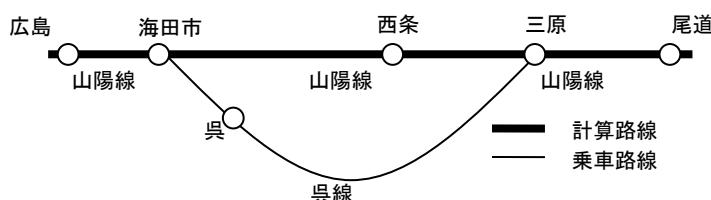
	神戸
→ 岩徳線	<b>岩国</b>
	櫛ヶ浜
→ 山陽線	下関

置換処理(1/3) : 乗車路線(1線)計算路線(1線)に限る実装(a69list is 1 / t\_rule69 is 1)

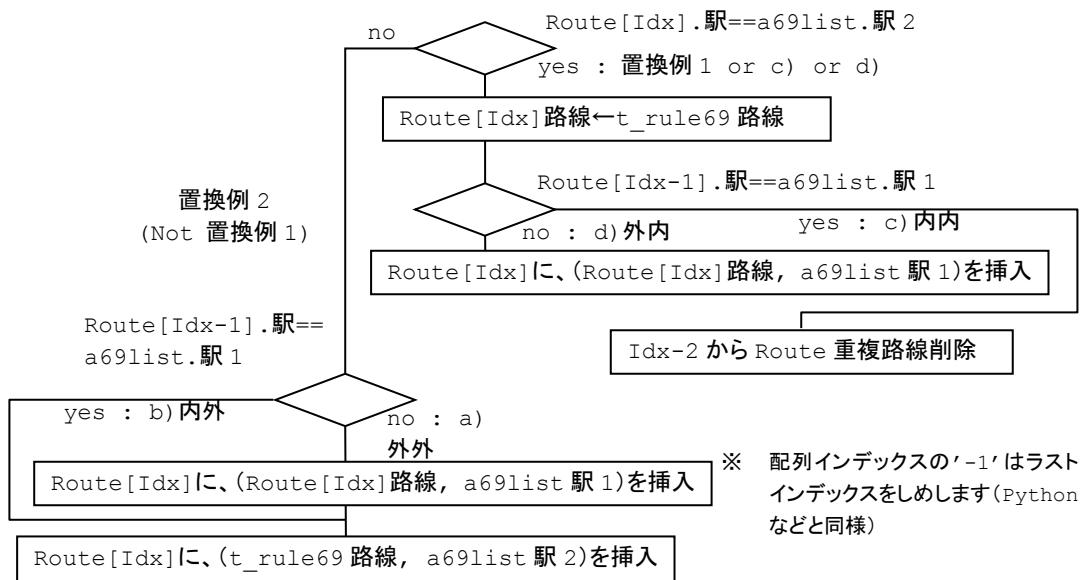
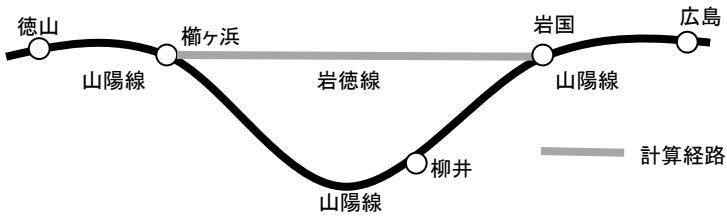
乗車駅(経路分岐駅かその外か)、路線(乗車 or 下車駅路線=乗車経路路線(置換例 2)／乗車 or

下車路線 ≠ 乗車経路路線(置換例 1))によって処理が異なる(上り下りとも同一処理で可)。

置換例 1



置換例 2



### 置換例 3-1 勝浦→東京

a69list

京葉	蘇我	東京
----	----	----

t\_rule69 recordset SQL

総武	東京	千葉
外房	千葉	蘇我

a), b) 非該当

c) 内→内

京葉	蘇我
外房	千葉
→ 組合せ	

外房線	蘇我
外房線	千葉
総武線	東京
→ 組合せ	

d) 外→内

外房線	蘇我
京葉線	東京
外房線	千葉
総武線	東京
→ 組合せ	

外房線	千葉
外房線	千葉
総武線	東京
→ 組合せ	

### 置換例 3-2 東京→勝浦

a69list

京葉	東京	蘇我
----	----	----

t\_rule69 recordset SQL

総武	東京	千葉
----	----	----

外房	千葉	蘇我
----	----	----

a) 非該当

b) 内→外

→	東京	東京
→	京葉	千葉
→	外房線	蘇我

→	東京	東京
→	総武線	千葉
→	外房線	蘇我

→	東京	東京
→	総武線	千葉
→	外房線	勝浦

c) 内→内

→	東京	東京
→	京葉	蘇我

→	東京	東京
→	総武線	千葉

→	東京	東京
→	外房線	蘇我

d) 外→内※ 外は無視できる(関係ない)例

→	JII崎	JII崎
→	東海道	東京
→	京葉	蘇我

→	東海道	東京
→	総武線	千葉

→	外房線	蘇我
---	-----	----

#### 置換処理(2/3) :乗車経路(1線)計算経路(2線)に限る実装(a69list is 1 / t\_rule69 is 2)

```
if a69list 駅 1==t_rule69[0]駅 1
    下り
else
    上り
endif
if 下り
    Route[Idx]線←t_rule69[1]線
    Insert to (t_rule69[0]線, t_rule69[0].駅 2) at Route[Idx]
else
    Route[Idx]線←t_rule69[0]線
    Insert to (t_rule69[1]線, t_rule69[1].駅 1) at Route[Idx]
endif

~重複路線を削除(Idx-2~)
```

#### 置換例 4-1 大阪→直江津

a69list

東海道線	山科	米原
北陸線	米原	近江塩津

t\_rule69 recordset SQL

湖西線	山科	近江塩津
-----	----	------

a) 外→外

→	大阪	大阪
→	東海道線	米原
→	北陸線	直江津

→	大阪	大阪
→	東海道線	山科
→	大阪	大阪
→	湖西線	近江塩津
→	大阪	大阪
→	北陸線	直江津

b) 内→外

	山科
東海道線	米原
→ 北陸線	直江津

⇒ →

	山科
湖西線	近江塩津
北陸線	直江津

c) 内→内

	山科
東海道線	米原
→ 北陸線	近江塩津

⇒ →

	山科
湖西線	近江塩津

d) 外→内

	大阪
東海道線	米原
→ 北陸線	近江塩津

⇒ →

	大阪
東海道線	山科
湖西線	近江塩津

#### 置換例 4-2 直江津→大阪

a69list

北陸線	近江塩津	米原
東海道線	米原	山科

t\_rule69 recordset SQL

湖西線	山科	近江塩津
-----	----	------

a) 外→外

	直江津
北陸線	米原
→ 東海道線	大阪

⇒ →

	直江津
北陸線	近江塩津
湖西線	山科
→ 東海道線	大阪

追加挿入(米原!=近江塩津なので、北陸線、米原→近江塩津へ置換)、湖西線、山科を追加

b) 内→外

	近江塩津
北陸線	米原
→ 東海道線	大阪

⇒ →

	近江塩津
湖西線	山科
東海道線	大阪

c) 内→内

	近江塩津
北陸線	米原
→ 東海道線	山科

⇒ →

	近江塩津
湖西線	山科

d) 外→内

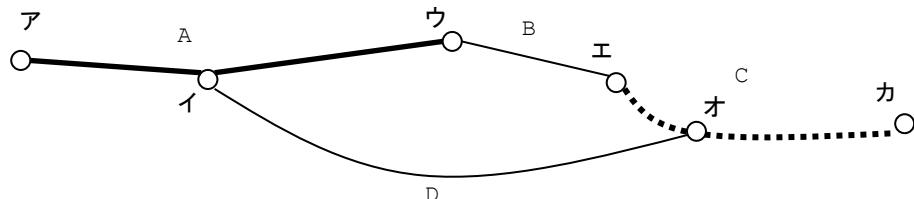
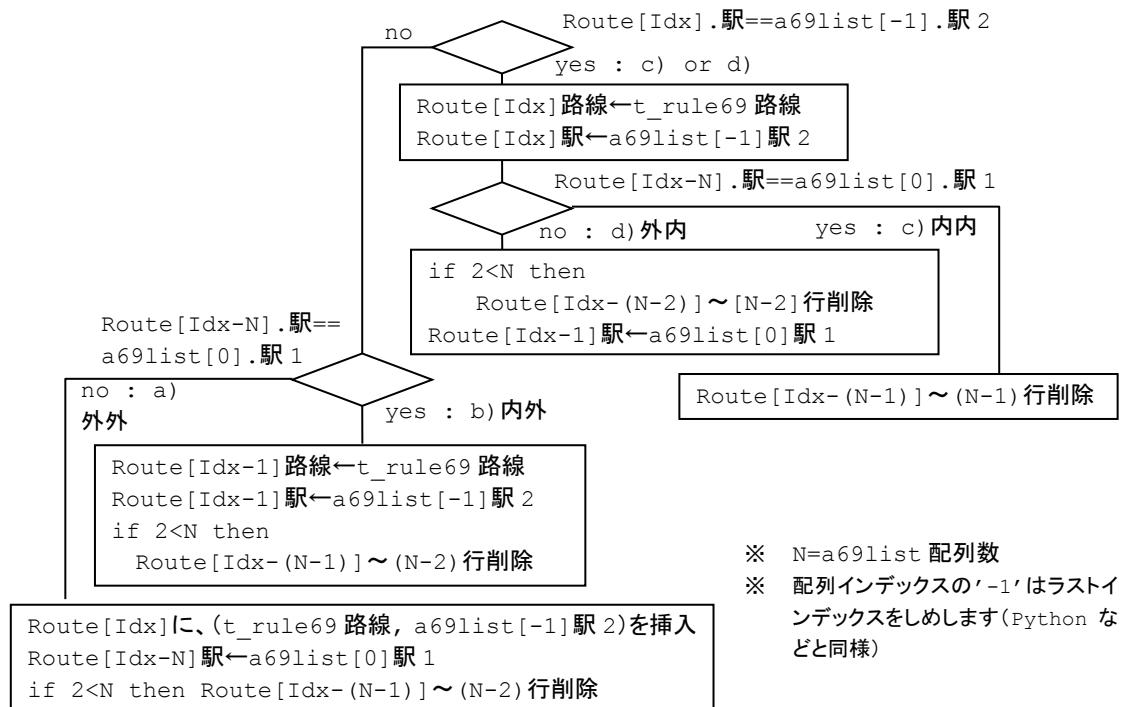
	直江津
北陸線	米原
→ 東海道線	山科

⇒ →

	直江津
北陸線	近江塩津
湖西線	山科

※ 米原までの場合、成立しない(HCP チャートの 2 段目のループで抜ける)

### 置換処理(3/3) : 乗車経路(N線)計算経路(1線)の実装(a69list is N / t\_rule69 is 1)



a69list

A	イ	ウ
B	ウ	工
C	工	才

t\_rule69

D	イ	才
---	---	---

	ア
A	ウ
B	工
C	才

→

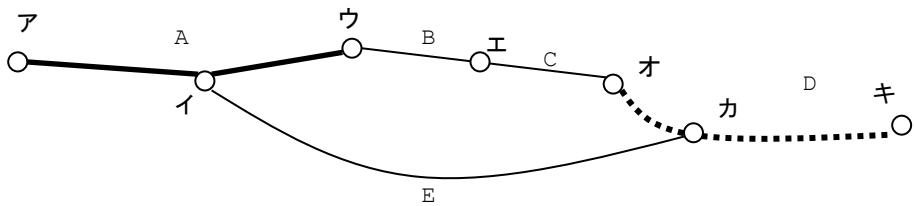
	ア
A	イ
B	工
C	才

	ア
A	ウ
B	工
C	才

→

	ア
A	イ
B	工
C	才

→



a69list

A	イ	ウ
B	ウ	エ
C	エ	オ
D	オ	力

t\_rule69

E	カ	イ
---	---	---

	ア
A	ウ
B	エ
C	オ
D	キ

	ア
A	イ
E	カ
D	キ

	ア
A	ウ
B	エ
C	オ
D	キ

	ア
A	イ
B	エ
C	オ
E	カ
D	キ

### 重複経路除去アルゴリズム(文字列の実装例)

```
#include <stdio.h>
#include <string.h>

void removeAtIndex(char* s, int pos)
{
    int len = strlen(s);
    if (pos < len) {
        strcpy(s + pos, s + pos + 1);
    }
}

void dupremov(char* s)
{
    int last = strlen(s) - 1;
    int idx = 0;

    while (idx < last) {
        if (s[idx] == s[idx + 1]) {
            removeAtIndex(s, idx);
            last--;
        } else {
            idx++;
        }
    }
}
```

```
    }

int main(int argc, char** argv)
{
    char buf[128];

    strcpy(buf, *++argv);
    dupremov(buf);
    printf("%s -> %s\n", buf, *argv);
    return 0;
}
```

## ■ 特定都市内駅の発着 86,87 条

特定都市内駅の発着時、中心駅と定められた駅から(まで)200km 以上は、中心駅から(まで)計算する。

ただし、特定都区市内から目的駅までの運賃が中心駅から 200km に達する最初の駅までの運賃と、実際に乗車する発駅～目的駅が 200km に満たない場合で、特定都区市内中心駅から目的駅までの運賃計算キロが大きい場合、前者の運賃を採用する(基 114)。すなわち、例として、運賃計算キロが横浜市内の中心駅である横浜駅から 200km に達した最初の駅 A までの運賃と、横浜駅から 200km に達しない駅で、本郷台からの運賃計算キロが横浜～駅 A までの運賃を越える場合、駅 A の運賃を採用する。

また、特定都区市内駅からいったんはずれて、再度、発駅(または着駅)と同一の特定都区市内駅を通過する場合は適用しないことができる(適用しない)。

200km／100km 以上は、営業キロである。(換算キロではない)

### 実装概要

~~規則 69 条の後におこなってから、本適用をおこない、さらに規則 69 条の適用をおこなう必要がある。~~  
~~(埼京線—浮島船戸、北赤羽着発)~~

本適用は、経路指定時、自動計算はしない。「都区市内発着」ボタン\*1 をクリックすることにより、適用する。再度「都区市内発着」ボタンをクリックすると

\*1 …「北九州市内発」／「単駅発」、「北九州市内着」／「単駅着」のチェックボタンが選択可能になる。

本適用後、69 条を適用し、201km 判定をおこない、未適用の場合、さらに規則 87 条で再スキャンして 100 km 判定をおこなう。

200 km／100 km 判定は、86 条、69 条適用後におこなう

### データ定義

テーブル t\_station

city_flg		
BCCYAMETE	山手線内駅	1 ビット
BCCITYCT	中心駅	1 ビット
BCCITYNO	都区市内番号	4 ビット

テーブル: rule86 は脱出(進入)路線への経路が中心駅からその路線への乗換え駅と路線を定義する。

以下の例 1 は、常磐線は、東京一東北線一日暮里一常磐線であり、例 2 は、東京一東北線であることをしめす。

一部大阪市内では例外があつて 2 線以内に到達しない例があり、2 レコードにまたがる例がある(「おおさか東線」の例)。この場合、2 レコード目の city\_id の bit8 から上位にレコード番号(2 レコード目=1)を表すため以下のクエリーとなる。

```
select * from t_rule86 where (city_id & 255)=?
```

テーブル t_rule86				
列名	line_id1	station_id	line_id2	city_id
説明	脱出(進入)路線	最初の乗換駅	最初の路線	都区市内番号
例 1 (2 線の例)	常磐線	日暮里	東北線	1
例 2 (1 線の例)	東北	東京	東北線	1

### 86 条／87 条適用パターン

	適用パターン	
A		適用
B		対象外
C		対象外
D		適用
E		適用
F		対象外(○内最短経路)
G		対象外
H		対象外

## 規則 86 条

### 1. 東京都区内

東京都区内(東京)	
東京	長野・北陸新幹線
東京	上越新幹線
東京	東北新幹線

品川	東海道新幹線
西荻窓	中央東線
西大井	東海道線(品川-鶴見)
蒲田	東海道線
葛西臨海公園	京葉線
小岩	総武線
金町	常磐線
浮間舟渡	東北線(赤羽-大宮)(埼京線)
赤羽	東北線



t_rurle86			
line id 1	station id	line id2	city id
常磐線	日暮里	東北線	1
中央東線	東京	中央東線	1
東海道線(西大井経由)	品川	東海道線	1
東海道線	東京	東海道線	1
京葉線	東京	京葉線	1
総武線	東京	総武線	1
東北線(埼京線)	赤羽	東北線	1
東北線	東京	東北線	1
長野・北陸新幹線	東京	長野・北陸新幹線	1
上越新幹線	東京	上越新幹線	1
東北新幹線	東京	東北新幹線	1
東海道新幹線	東京	東海道新幹線	1

以降の凡例

用語	DB フィールド名
脱出路線／進入路線	line_id1
乗換駅	station_id
中心駅発着路線	line_id2

<パターン：脱出≠>

指定経路

	蒲田
東海道線	品川
山手線	代々木
中央東線	御茶ノ水
総武線	秋葉原
東北線	田端
山手線	池袋
赤羽線	赤羽
<b>埼京線</b>	<b>大宮</b>
高崎線	高崎
上越線	宮内
信越線	新津
羽越線	秋田

規則 86 条適用

	都区内 (東京)
<b>東北線</b>	<b>赤羽</b>
埼京線	<b>大宮</b>
高崎線	高崎
上越線	宮内
信越線	新津
羽越線	秋田

規則 69 条適用

	都区内 (東京)
東北線	大宮
高崎線	高崎
上越線	宮内
信越線	新津
羽越線	秋田

脱出路線(埼京線) ≠ 中心駅発着路線(東北線)なので脱出路線の行の前に行挿入しその行を中心駅発着路線(東北線)と乗換駅(赤羽)とし、その前の行の駅を中心駅(東京)(flg.BCCITYNO=cityflg.BCCITYNO の都区市内番号を設定)にする。それ以前の行の路線は削除する。(2段目列)

その後、規則 69 条適用(3段目列)

<パターン：进入≠>

指定経路

	秋田
羽越線	新津
信越線	宮内
上越線	高崎
高崎線	大宮
<b>埼京線</b>	<b>赤羽</b>
赤羽線	池袋
山手線	田端
東北線	秋葉原
総武線	御茶ノ水
中央東線	代々木
山手線	品川
東海道線	蒲田

規則 86 条適用後経路

	秋田
羽越線	新津
信越線	宮内
上越線	高崎
高崎線	大宮
<b>埼京線</b>	<b>赤羽</b>
東北線	東京

規則 69 条適用

	秋田
羽越線	新津
信越線	宮内
上越線	高崎
高崎線	大宮
<b>埼京線</b>	<b>赤羽</b>
東北線	東京

进入路線(埼京線) ≠ 中心駅発着路線(東北線)なので、その直後を中心駅発着路線(東北線)と中心駅(東京)(東京都区内)とする(flg.BCCITYNO=cityflg.BCCITYNO の都区市内番号を設定)。进入路線の駅は、乗換駅(赤羽)に置換。进入路線の行+2 以降の行は削除する。(2段目列)

その後、規則 69 条適用(3 段目列)

<パターン:脱出=>

	蒲田
東海道線	品川
山手線	代々木
中央東線	御茶ノ水
総武線	秋葉原
東北線	盛岡

	東京
東北線	盛岡

flg.BCCITYNO= 東京都区内

脱出路線(東北線)=中心駅発着路線(東北線)なので脱出路線の行-1の行の駅を中心駅

(東京)(flg.BCCITYNO=cityflg.BCCITYNO の都区市内番号を設定)にし、路線は削除。

それ以前(脱出路線の行-2 以前)の行は削除する。(2 段目列)

その後、規則 69 条適用(3 段目列)

<パターン:进入=>

	盛岡
東北線	秋葉原
総武線	御茶ノ水
中央東線	代々木
山手線	品川
東海道線	蒲田

	盛岡
東北線	東京

flg.BCCITYNO= 東京都区内

进入路線(東北線)=中心駅発着路線(東北線)なので、进入路線の行の駅を中心駅(東京駅)(東京都区内)とする(flg.BCCITYNO=cityflg.BCCITYNO の都区市内番号を設定)。それ以降(进入路線の行+1 以降)の行は削除する。

	蒲田
東海道線	東京
東北線	盛岡

	東京
東北線	盛岡

(同上)

	赤羽
東北線	盛岡

	東京
東北線	盛岡

(同上)

指定経路

	赤羽
埼京線	大宮
東北線	盛岡

規則 86 条適用後経路

	東京
東北線	赤羽
埼京線	大宮
東北線	盛岡

規則 69 条適用

	東京
東北線	盛岡

<パターン:脱出≠>

	蒲田
東海道線	東京
中央東線	西国分寺
武蔵野線	武蔵浦和
埼京線	大宮
東北線	盛岡

	東京
中央東線	西国分寺
武蔵野線	武蔵浦和
埼京線	大宮
東北線	盛岡

<パターン:脱出=>

	浮間舟渡
埼京線	赤羽
赤羽線	池袋
山手線	新宿
<b>中央東線</b>	<b>松本</b>

	東京
中央東線	松本

<パターン:脱出=>

指定経路

	浮間舟渡
<b>埼京線</b>	<b>大宮</b>
東北線	盛岡

86条適用

	東京
<b>東北線</b>	<b>赤羽</b>
埼京線	大宮
東北線	盛岡

69条適用

	東京
東北線	赤羽
東北線	盛岡

重複最適化

	東京
東北線	盛岡

<パターン:脱出≠>

指定経路

	盛岡
東北線	大宮
<b>埼京線</b>	<b>浮間舟渡</b>

86条適用

	盛岡
東北線	大宮
埼京線	<b>赤羽</b>
東北線	東京

69条適用

	盛岡
東北線	赤羽
東北線	東京

重複最適化

	盛岡
東北線	東京

<パターン:進入≠>

(2段目列)

その後、規則 69 条適用(3段目列)

69 条適用(3段目列)

重複路線削除(4段目列)

	浮間舟渡
埼京線	赤羽
<b>東北線</b>	<b>盛岡</b>

	東京
東北線	盛岡

<パターン:脱出=>

	浮間舟渡
<b>埼京線</b>	<b>武藏浦和</b>
武藏野線	南浦和
東北線	盛岡

	都区内
<b>東北線</b>	<b>赤羽</b>
埼京線	武藏浦和
武藏野線	南浦和
東北線	盛岡

<パターン:脱出≠>

	浮間舟渡
<b>埼京線</b>	<b>武藏浦和</b>
武藏野線	新松戸
常磐線	岩沼
東北線	盛岡

	東京
<b>東北線</b>	<b>赤羽</b>
埼京線	武藏浦和
武藏野線	新松戸
常磐線	岩沼
東北線	盛岡

<パターン:脱出≠>

	板橋
--	----

	東京
--	----

赤羽線(埼京線)	赤羽
東北線(尾久経由)	日暮里
東北線	東京
<b>東海道新幹線</b>	<b>名古屋</b>

東海道新幹線	名古屋
--------	-----

<パターン: 脱出=>

	板橋
赤羽線(埼京線)	赤羽
東北線(尾久経由)	日暮里
東北線	秋葉原
総武線(御茶ノ水-錦糸町)	御茶ノ水
中央東線	代々木
山手線	品川
<b>東海道新幹線</b>	<b>名古屋</b>

	東京
東海道新幹線	名古屋

<パターン: 脱出=>

	蒲田
<b>東海道線</b>	<b>川崎</b>
南武線	立川
中央東線	松本

	東京
東海道線	川崎
南武線	立川
中央東線	松本

<パターン: 脱出=>

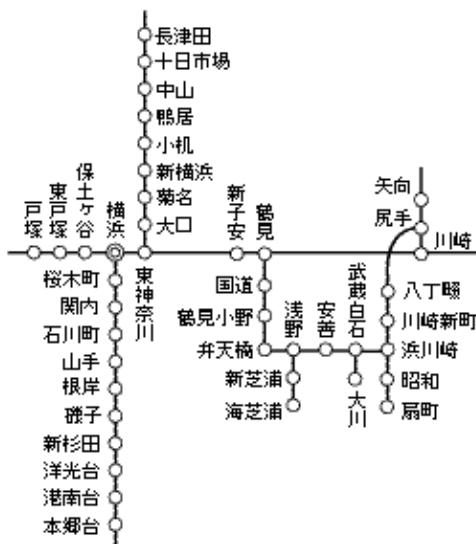
	西大井
<b>東海道線(西大井経由)</b>	<b>武藏小杉</b>
南武線	府中本町
武蔵野線	新松戸
常磐線	岩沼

	東京
東海道線	品川
東海道線(西大井経由)	武藏小杉
南武線	府中本町
武蔵野線	新松戸
常磐線	岩沼

<パターン: 脱出≠>

## 2. 横浜市内

横浜市内(横浜)	
長津田	横浜線
戸塚	東海道線
本郷台	根岸線
川崎	東海道線
矢向	南武線

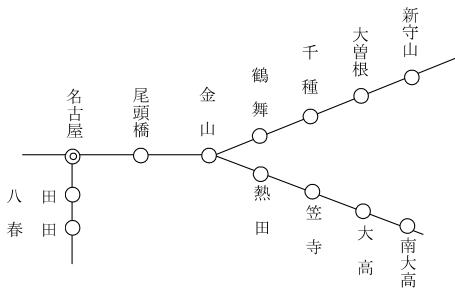


t_rurle86			
line id1	station id	line id2	city id
横浜線	東神奈川	東海道線	2
東海道線	横浜	東海道線	2
根岸線	横浜	根岸線	2
南武線	川崎	東海道線	2
東海道新幹線	新横浜	東海道新幹線	2

小海線	海尻
中央東線	小淵沢
横浜線	八王子
横浜線	長津田

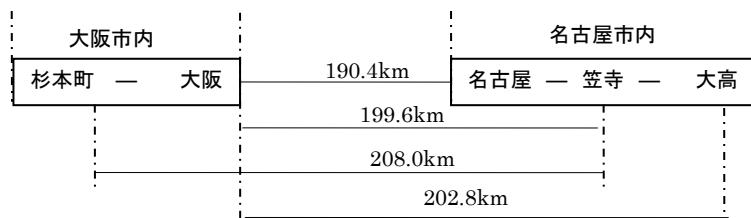

## 3. 名古屋市内

名古屋市内(名古屋)	
名古屋	東海道線、東海道新幹線
春田	関西線
南大高	東海道線
新守山	中央西線



t_rurle86			
line_id1	station_id	line_id2	city_id
東海道新幹線	名古屋	東海道新幹線	3
東海道線	名古屋	東海道線	3
中央西線	金山	東海道線	3
関西線	名古屋	関西線	3

大高—(東海道線、大阪環状線)—杉本町は、  
 大阪—名古屋が、190.4km であるが、杉本町—名古屋 208.0km、大阪—大高 202.8km で  
 あるので、大阪市内—大高または、杉本町—名古屋市内のいずれかで OK.  
 ただし、デフォルトでは申し出のない場合、着駅を特定都市内で発券される。  
 ただし、名古屋—杉本町と指定した場合、名古屋市内—杉本町となる



発	着	発券		
大高	杉本町	大高—大阪市内	202.8km	名古屋—大阪 190.4km で満たないので 名古屋—杉本町なら満たす 大高—大阪市内も満たすので着駅優先で後者採用
笠寺	杉本町	名古屋市内—杉本町	208.0km	名古屋—大阪 190.4km で満たない 笠寺—大阪 199.6km で満たないので、 名古屋市内—杉本町となる

```
A s---c-----c---g
B c---s-----c---g
C s---c-----g---c
D c---s-----g---c
```

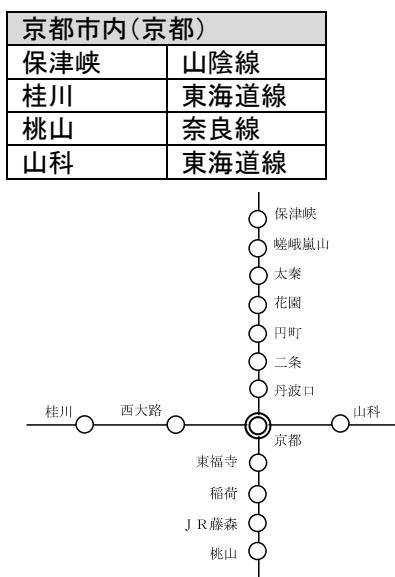
Start, Center, Goal

```

if c - c > 200
  c - c
else if s - c > 200
  s - c          //A
else if c - g > 200
  c - g          //D
else

```

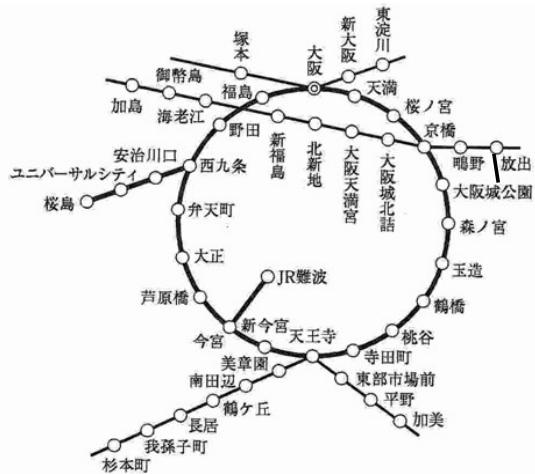
#### 4. 京都市内



t_rurle86			
line_id1	station_id	line_id2	city_id
東海道新幹線	京都	東海道新幹線	4
東海道線	京都	東海道線	4
山陰線	京都	山陰線	4
奈良線	京都	奈良線	4
湖西線	山科	東海道線	4

#### 5. 大阪市内

大阪市内(大阪)	
塙本	東海道線
加島	JR 東西線
杉本町	阪和線
加美	関西線
放出	片町線
東淀川	東海道線



t_rurle86			
line_id1	station_id	line_id2	city_id
東海道新幹線	新大阪	東海道新幹線	5
東海道線	大阪	東海道線	5
山陽新幹線	新大阪	山陽新幹線	5
おおさか東線	放出	片町線	5
おおさか東線	京橋	大阪環状線	261 (0x105)
JR東西線	京橋	大阪環状線	5
片町線	京橋	大阪環状線	5
関西線	天王寺	大阪環状線	5
阪和線	天王寺	大阪環状線	5

※ おおさか東線は 2 レコード定義で、city\_id>0xff は 2 レコード目を指す。

したがって、脱出路線が「おおさか東線」の場合、以下となる

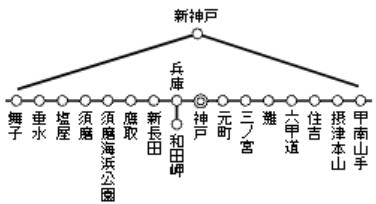
※ (脱出のパターン:=はあり得ない)

	放出		大阪
おおさか東線	久宝寺	大阪環状線	京橋
関西線	王子	片町線	放出
和歌山線	和歌山	おおさか東線	久宝寺
紀勢線	亀山	関西線	和歌山
関西線	名古屋	紀勢線	亀山
		関西線	名古屋

	名古屋		名古屋
関西線	亀山	関西線	亀山
紀勢線	和歌山	紀勢線	和歌山
和歌山線	王子	和歌山線	王子
関西線	久宝寺	関西線	久宝寺
おおさか東線	放出	おおさか東線	放出
		片町線	京橋
		大阪環状線	大阪

## 6. 神戸市内

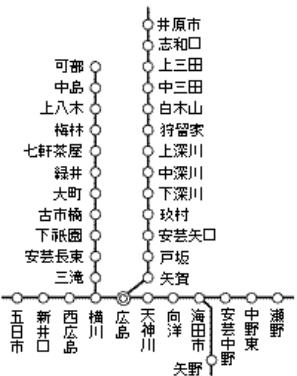
神戸市内(神戸)	
舞子	山陽線
甲南山手	東海道線



t_rurle86			
line id1	station id	line id2	city id
山陽線	神戸	山陽線	6
山陽新幹線	新神戸	山陽新幹線	6
東海道線	神戸	東海道線	6

## 7. 広島市内

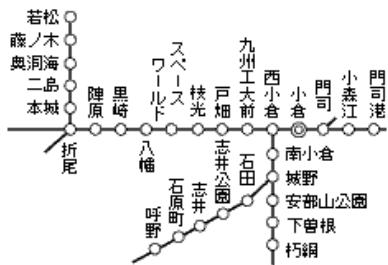
広島市内(広島)	
井原市	芸備線
五日市	山陽線
矢野	呉線
瀬野	山陽線



t_rurle86			
line id1	station id	line id2	city id
山陽線	広島	山陽線	7
山陽新幹線	広島	山陽新幹線	7
芸備線	広島	芸備線	7
可部線	横川	山陽線	7
吳線	海田市	山陽線	7

## 8. 北九州市内

北九州市内(小倉)	
折尾	筑豊線、鹿児島線
呼野	日田彦山線
朽網	日豊線
門司	鹿児島線



t_rurle86			
line id1	station id	line id2	city id
鹿児島線	小倉	鹿児島線	8
日豊線	小倉	日豊線	8
日田彦山線	城野	日豊線	8
筑豊線	折尾	鹿児島線	8

## 9. 福岡市内

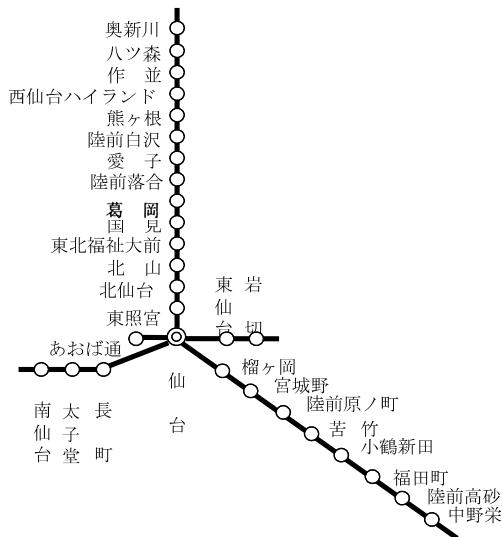
福岡市内(博多)	
博多	博多南線
博多	九州新幹線
博多	山陽新幹線
南福岡	鹿児島線
土井	香椎線
福工大前	鹿児島線



t_rurle86			
line id1	station id	line id2	city id
鹿児島線	博多	鹿児島線	9
篠栗線	吉塚	鹿児島線	9
香椎線	香椎	鹿児島線	9
九州新幹線	博多	九州新幹線	9
山陽新幹線	博多	山陽新幹線	9

## 10. 仙台市内

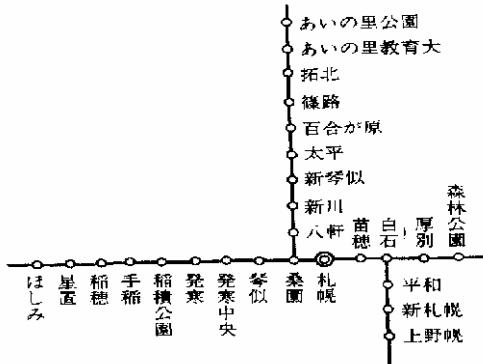
仙台市内(仙台)	
奥新川	仙山線
南仙台	東北線
中野栄	仙石線
岩切	東北線



t_rurle86			
line_id1	station_id	line_id2	city_id
東北線	仙台	東北線	10
東北新幹線	仙台	東北新幹線	10
仙石線	仙台	仙石線	10
仙山線	仙台	仙山線	10

## 11. 札幌市内

札幌市内(札幌)	
あいの里公園	札沼線
ほしみ	函館線
森林公園	函館線
上野幌	千歳線



t_rurle86			
line_id1	station_id	line_id2	city_id
函館線	札幌	函館線	11
札沼線	桑園	函館線	11
千歳線	白石	函館線	11

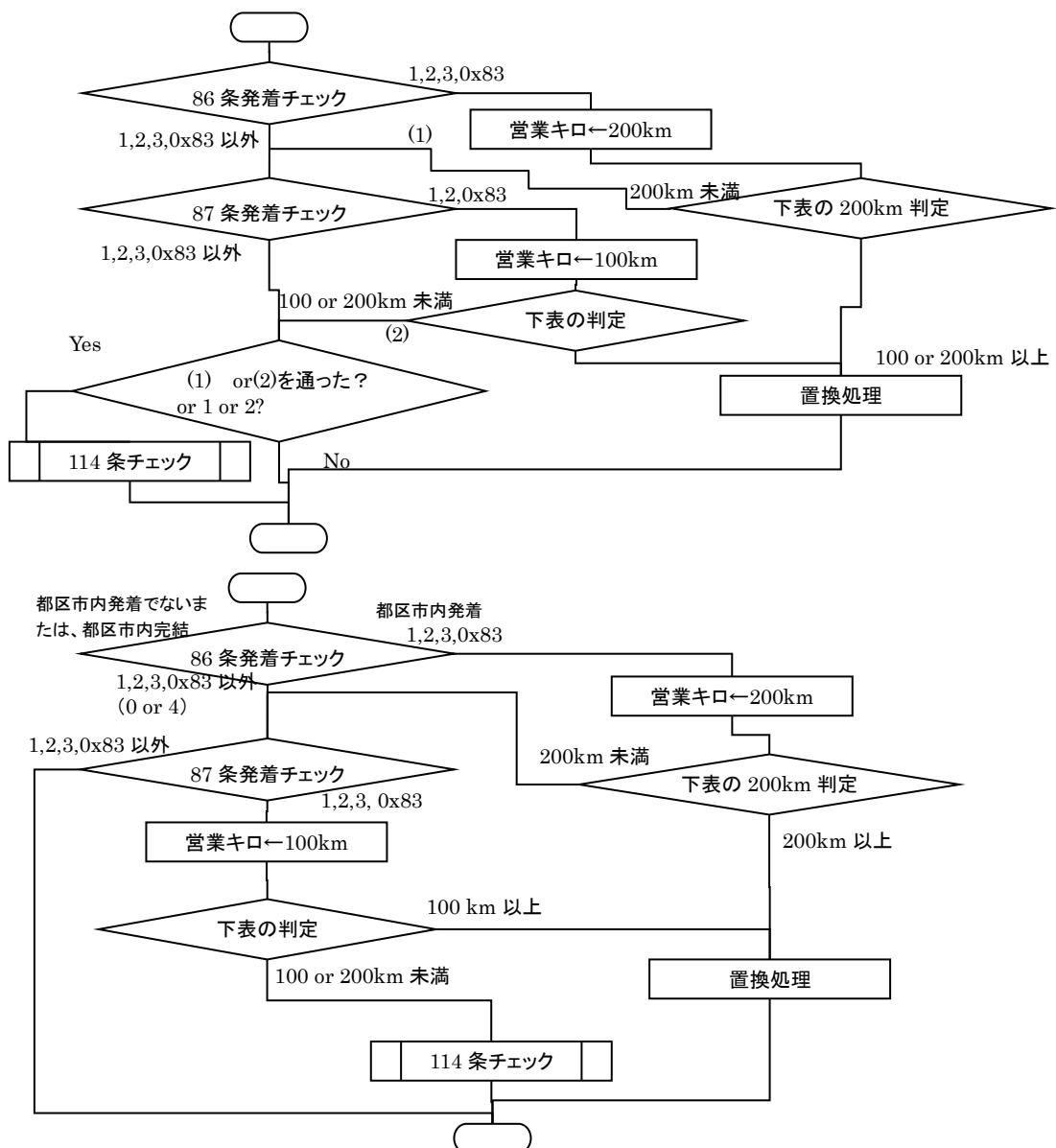
## 実装詳細

本適用後、69 条を適用後、201km 判定をおこない、未適用の場合、さらに規則 87 条で再スキャンし

て 100 km 判定をおこなう。

200 km / 100 km 判定は、86 条、69 条適用後におこなう

### 86/87 条判定フロー



戻値	発着チェック結果	判定処理
0	発駅、着駅とも単駅(通常駅)	—
1	発駅のみ特定都市内駅	発駅を特定都区市内に一時的に置き換えて営業キロが満たしているか判定
2	着駅のみ特定都市内駅	着駅を特定都区市内に一時的に置き換えて営業キロが満たしているか判定
3	発駅、着駅とも特定都市内駅	発着駅を特定都区市内に一時的に置き換えて営業キロが満たしているか判定
0x83	発駅、着駅とも同一特定都市内駅	発着駅を特定都区市内に一時的に置き換えて営業キロが満たしているか判定
4	全駅同一特定都市内駅	—

- 200km 越えていれば 86 条なので、86 条駅 ⇄ 87 条駅はありえない。

### ● 特定都市内駅の発着(規則 86、87 条の適用)チェック

#### 86 条チェック

```
int checkOfSpecificStartAndEndCityTerminal(route)
    route : RouteItem 配列
    return 0: 発駅、着駅とも単駅(通常駅)
            1: 発駅のみ特定都市内駅
            2: 着駅のみ特定都市内駅
            3: 発駅、着駅ともそれぞれ別々の特定都市内駅
            0x83: 発駅、着駅とも同一特定都市区内(O の字)
            4: 全駅同一特定都市内駅
```

全ルート配列を対象に特定都区市内発着チェックをおこなう(引数は Route 配列)

ルート配列の駅(=発駅、分岐駅、着駅)のみをチェックすることにより、86 条適用を検出するが、以下のケースで 86 条文のただし書きを見落とすことになるため、これを考慮する実装にする必要がある。

- [九] 山陽新幹線
- [福] 鹿児島線
- [神] 山陽新幹線
- [札] 函館線
- [仙] 東北新幹線、東北線
- [広] 山陽新幹線、山陽線
- [阪] 東海道線
- [京] 東海道線
- [名] 東海道線
- [浜] 東海道線

例えば、[浜]石川町発で、大船から、東海道線で東京へ行く場合、経路配列だけでは検出できない。

逆に、東京から東海道線で大船から、[浜]石川町着の場合も、検出できない([浜]とはならず、単駅でなければならない)

以下 1 ルートパターンについて

○を 86 適用駅、両端が分岐駅(ルート配列に含まれる駅)とする。

-----	都区市内の外
0000000000	都区市内の中
0-----	都区市内を出た
-----0	都区市内に入った
0-----0	このパターンはなし
-0-----0-	このパターンのみ、ルート配列の駅のみでは判定できない(上記の「ただし書きを見落とす例」)

## 87 条チェック

```
int checkOfSpecificStartAndEndCityTerminalTM(route)
{
    route : RouteItem 配列
    return 0: 発駅、着駅とも単駅(通常駅)
            1: 発駅のみ特定都市内駅
            2: 着駅のみ特定都市内駅
            3: 発駅、着駅ともそれぞれ別々の特定都市内駅(ありえない)
            0x83: 発駅、着駅とも同一特定都市区内(0 の字)
            4: 全駅同一特定都市内駅
```

全ルート配列を対象に東京山手線内発着チェックをおこなう(引数は Route 配列)

TM は、Tokyo Metro の略

## ● 置換処理

先述の「<パターン:脱出≠>」、「<パターン:進入≠>」、「<パターン:脱出=>」、「<パターン:進入=>」(79~80 ページ)を参照

次ページに 86 条チェックのフローをしめす。

87 条チェックは、86 条チェックの処理と同等で簡素化されたものとなる。

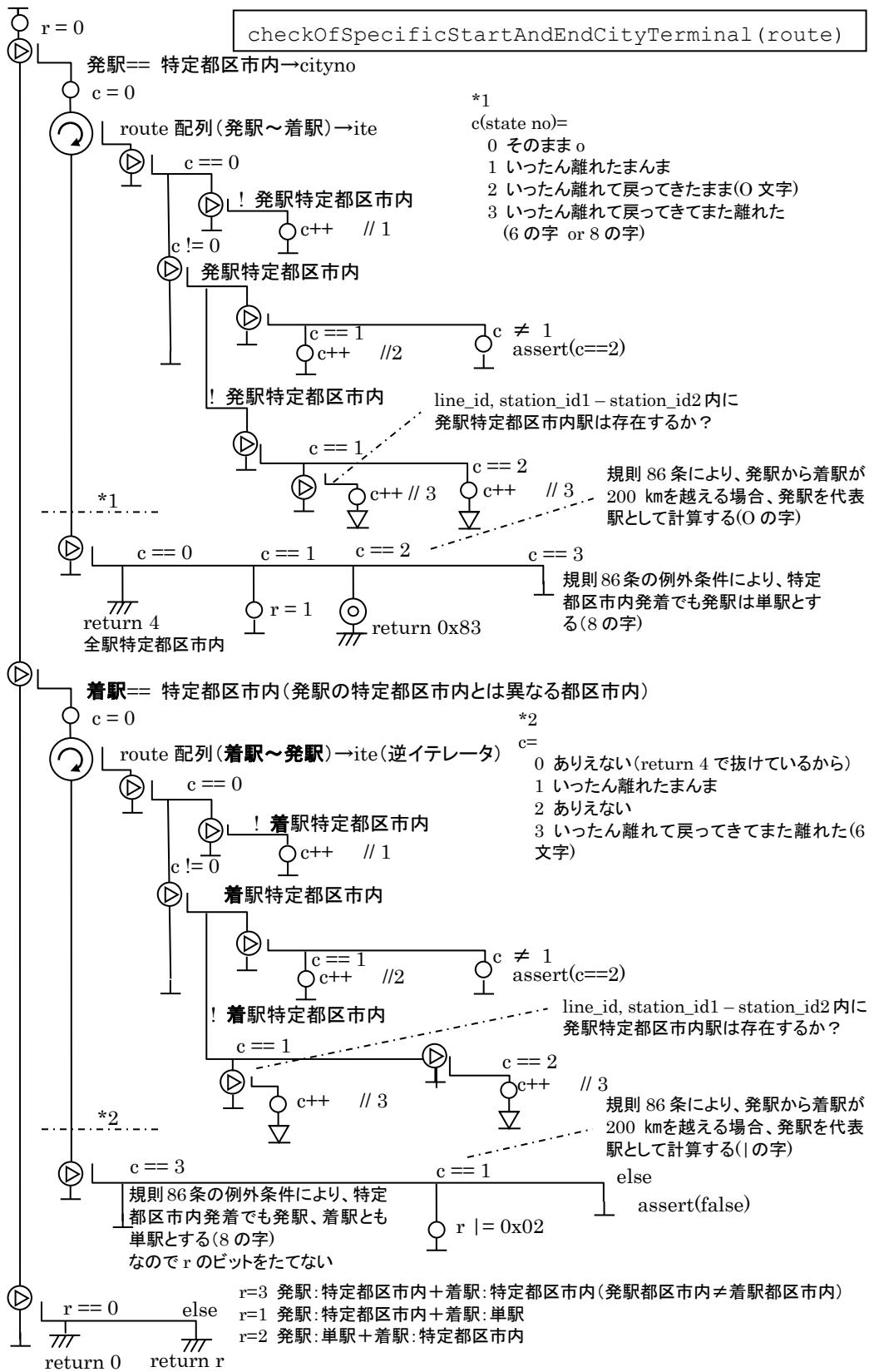
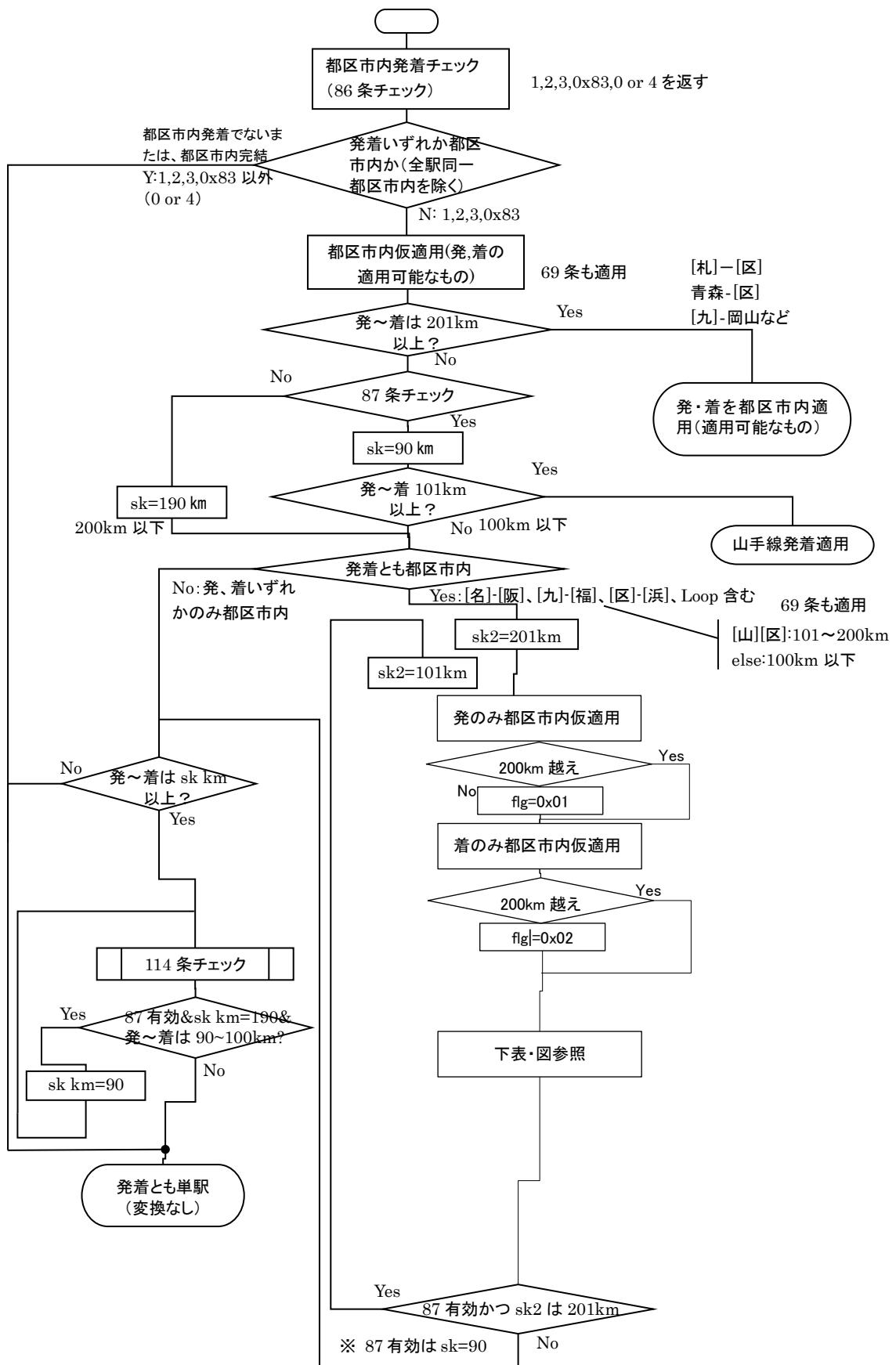


図 5

○の字は同一特定都区市内駅の発着(発駅に戻る)のほかに、発駅も着駅も(発駅と着駅の異なる)特定都区内である場合も含む(都区内-都区内や都区内-札幌市内)。

```
bool Route::checkOfRuleSpecificCoreLine(int dis_cityflag, int* rule114)
dis_cityflag=適用フラグ
最新(2013-Jan)フロー(ルート:概要フロー)
```



↑ 大阪—名古屋間の(あ)の適用可否は?、88条、69条チェックはどのタイミングでいれる?

単駅指定の場合の扱い

距離を算定するため、86条適用前に 69変換する。86適用後は不要である。

86 通過パターンは？？？

70 条を一番先に変換

69 条は一番先に変換しておく。

その後、86 をやる

69 条→70 条→86 条？

70 条→69 条→86 条？ ←

杉本町 大阪 名古屋 大高

~~発駅、着駅とも都区市内で、発着 200km 以下の時~~

発	着	201 km 以上	
=	=	θ	
=	=	=	[区]—[浜]など
=	θ	θ	着を適用
=	θ	=	発を適用してみる↓
θ	=	θ	発を適用
θ	=	=	単駅発着
θ	θ	θ	[札]—[区]など
θ	θ	=	

○ 適用、非適用

名古屋、金山、熱田、笠寺、大高

杉本町、天王寺、大阪、東淀川

名古屋—大阪 190.4km

太字は MARS.exe デフォルト発券(実装目標も同様)

指定区間	単・単	単・	- 単	特・特	
大高—天王寺	213.5	大高—[阪] <b>202.8</b>	[名]・天王寺 201.1	*1	
天王寺-大高	213.5	天王寺-[名] <b>201.1</b>	[阪]-大高 202.8	*1	
大高—東淀川	198.3	大高—[阪] <b>202.8</b>	[名]-東淀川 185.9	*2	
笠寺—天王寺	210.3	笠寺-[阪] 199.6	[名]・天王寺 <b>201.1</b>	*2	
金山—東淀川	<b>189.2</b>	金山-[阪] 193.7	185.9	*2	
金山—杉本町	211.3	金山-[阪] 193.7	[名]・杉本町 <b>208.0</b>	*2	
杉本町-金山		杉本町-[名] <b>208.0</b>	[阪]-金山 193.7	*2	
笠寺—杉本町	217.2	笠寺-[阪] 199.6	[名]・杉本町 <b>208.0</b>	*2	
杉本町-笠寺		杉本町-[名] <b>208.0</b>	[阪]-笠寺 199.6	*2	

\*1 適用フラグ=0 または 1 の場合(初期値)、開始駅の lineId のビット 7=ON とする(発駅のみ特定都

区市内適用し、着駅を単駅にする)。

適用フラグ 2 の場合、ビット 7=Off、ビット 6=ON とする(着駅のみ特定都区市内適用し、発駅を単駅にする)。

\*2 ビット 6、7 は、Off にする(適用フラグは無効)

取り消し線は発券しない。

着を適用してみて 201km 以上なら、そのまま単駅-[都区市内]とする(天王寺一大高の例)

着を適用してみて 200km 以下(笠寺ー杉本町や天王寺[大阪])の場合、発を適用してみて 201km 以上なら、そのまま[都区市内]-単駅とする(笠寺ー杉本町や天王寺)、200km 以下なら単駅発着とする。

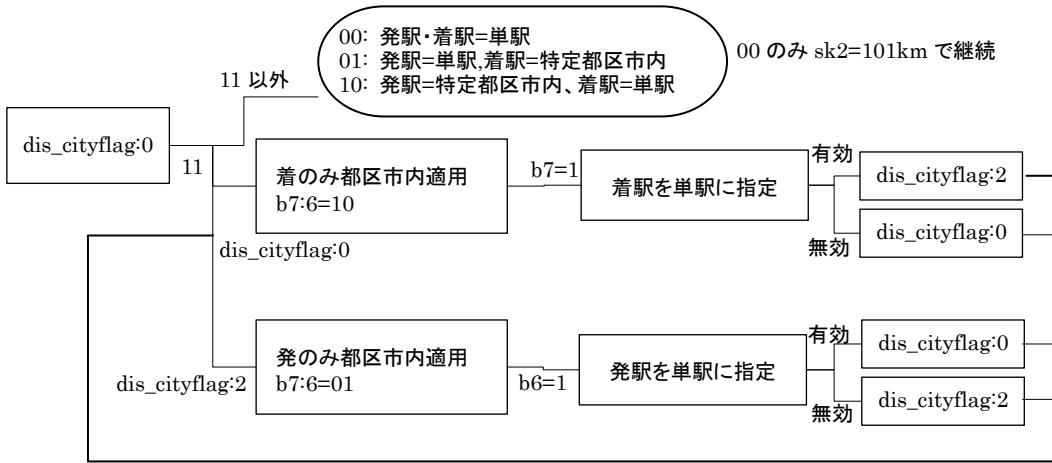
適用フラグについては、GUI 側で指定する。

「データ構造」-「RouteItem オブジェクト」の開始駅の lineId の説明も参照

入力	結果	出力		備考
dis_cityflag	flg	開始駅の lineId	処理	
0	00	b7:6=00	発駅／着駅=単駅	
	01	b7:6=00	発のみ都区市内適用	
	10	b7:6=00	着のみ都区市内適用	
	11	b7:6=10	着のみ都区市内適用	
1	00	b7:6=00	発駅／着駅=単駅	あり得ない
	01	b7:6=00	発のみ都区市内適用	あり得ない
	10	b7:6=00	着のみ都区市内適用	あり得ない
	11	b7:6=10	着のみ都区市内適用	
2	00	b7:6=00	発駅／着駅=単駅	あり得ない
	01	b7:6=00	発のみ都区市内適用	あり得ない
	10	b7:6=00	着のみ都区市内適用	あり得ない
	11	b7:6=01	発のみ都区市内適用	

運賃計算(表示)後、開始駅の lineId の b7:6=10 のとき、「着駅:単駅指定」を選択可となり、有効なとき、 dis\_cityflag を 2 で呼び出す(無効な時は、0 または 1 で呼び出す)

運賃計算(表示)後、開始駅の lineId の b7:6=01 のとき、「発駅:単駅指定」を選択可となり、有効なとき、 dis\_cityflag を 1(または 0)で呼び出す(無効な時は、2 で呼び出す)



## ■ 基準規定 114 条

特定都区市内にある駅と、その代表駅から営業キロ 200km 以下の区間にある駅との相互間についての鉄道の旅客運賃は、その関連する特定都区市内の代表駅から同一の方向および経路による営業キロが 200km を超える区間にある駅との普通旅客運賃に比較して、これよりも高額となる場合は、その同一の方向および経路による規則 86 条の規定を受ける最近の駅までの普通旅客運賃をもってこの区間の旅客運賃とすることができる。

(東京山手線発着の規則 87 条における 100km の規定についても同様)

### データ定義

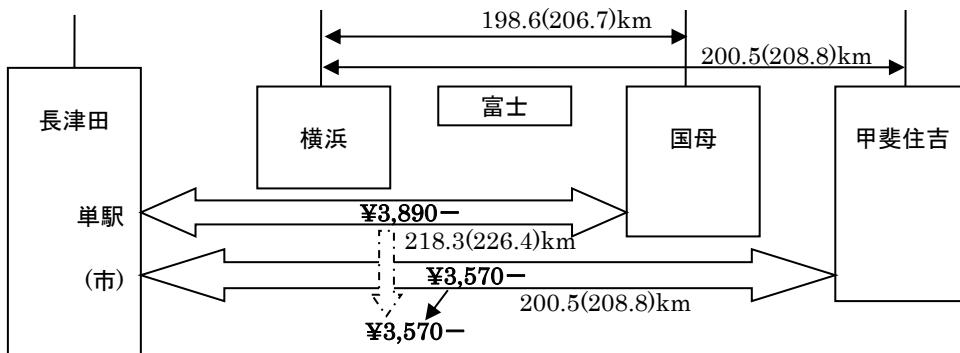
テーブル t_station		
city_flg		
BCYAMETE	山手線内駅	1 ビット
BCCITYCT	中心駅	1 ビット
BCCITYNO	都区市内番号	4 ビット
BCC114	都区市内中心駅から 100km、200km に達する最初の駅（114 条チェック用）(BCCITYNO ≠ 0 かつ、spe_route.BSRC114 ≠ 0 の路線でのみ有効)	1 ビット

テーブル t_lines		
specifit_flg		
BSRC114	00:無効 01:下り線で有効 10:上り線で有効	2 ビット
BSRD114	中心駅からの運賃計算キロ - 100km /	5 ビット

## &lt;適用される例&gt;

長津田から[東神奈川]、東海道線、[富士]、身延線経由

長津田	横浜	国母	甲斐住吉
19.7	0	198.6	200.5
	0	206.7	208.8
		¥3,570-	
		218.3	
		226.4	
		¥3,890	
			¥3,570-



長津田－国母 ¥3,890-(横浜－国母が 200km 以内なので長津田から計算)

長津田－甲斐住吉 ¥3,570-(横浜－甲斐住吉が 200km を越えるので、横浜から計算)

上記のとおり国母より遠い甲斐住吉の方が安いので、甲斐住吉までの料金と同一とする。

他に、井原市から広島を経て山陽本線上り方面など

A) 御茶ノ水－岩舟(代々木、原宿、田町、上野、土呂、小山経由)

123.6km／¥2,210-

B) 御茶ノ水－岩舟(秋葉原、東北線、両毛線経由:A 経路での乗車可能。但し 100km 未満なので途中下車ダメ)

98.8km／¥1,620-

C) 東京－岩舟 99.9km／¥1,620-

D) 東京－佐野 107.2km／¥1,890-

※ 東京から、岩舟の先の隣駅が佐野

A)は、東京(最短距離)岩舟が 100km 未満なので単駅計算で、¥2,210 であるが、岩舟の一つ先の佐野は、100km を越えるので山手線内発で、¥1,890-(D で計算)と安いため、¥1,890-として発券可能(マルスは非対応らしい)(新幹線は除外)

B)は、157 条 2 項(近郊区間)による発券で、通常発券される切符。ただし途中下車はできない(新幹線は除外)

C) は 100km 未満なので適用されない(参考)

D)は、100km 以上なので、山手線内着発が適用され、山手線内各駅から佐野までの切符はすべて東京駅発で計算される(新幹線は除外)

---

A)の経路で、御茶ノ水～岩舟を発券

(ア) 東京から岩船までは 100km 未満(99.9km:①)なので単駅計算→ ¥2,210

(イ) 近郊区間なので 157 条 2 項適用で、最短経路で計算→ ¥1,620

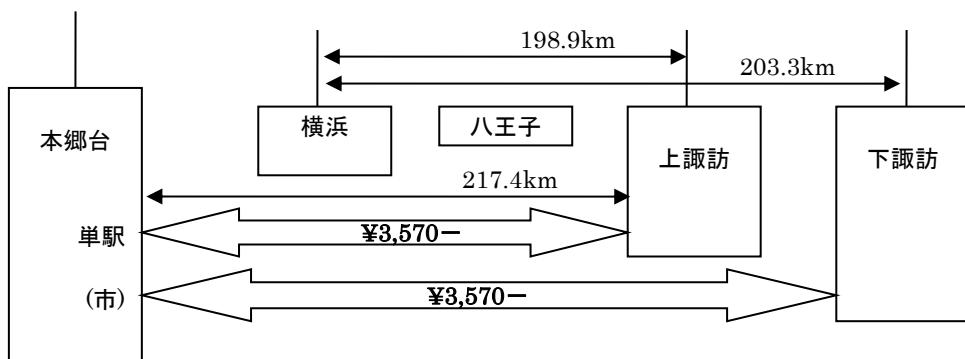
(ウ) 途中下車したいので 157 条条 2 項非適用で、→ ¥2,210

(エ) 東京-岩船が 99.9km と微妙なので(114 条候補)、岩舟の先の 100km に達する最初の駅(岩舟のさきの佐野)までの運賃を算出→ 107.2km／¥1,890-

(オ) ¥2,210 より安いので、114 条適用可能で A の経路は、¥1,890 で発券可能

---

<適用されない例>



横浜－(横浜線、[八王子]、中央東線経由)－上諏訪 198.9km ¥3,260-

横浜－(横浜線、[八王子]、中央東線経由)－下諏訪 203.3km ¥3,570-

本郷台－(横浜線、[八王子]、中央東線経由)－上諏訪 217.4km ¥3,570-

営業キロは逆転するが、料金は同一に納まっている。

## 実装詳細

「86／87 条判定フロー」(90 ページ)の処理内でおこなわれる。

以下の条件に合致したものについて 114 条かどうかを判定する

- 発着駅は、86 条、87 条候補だが、営業キロが、200km、100km に満たない為非該当で、営業キ

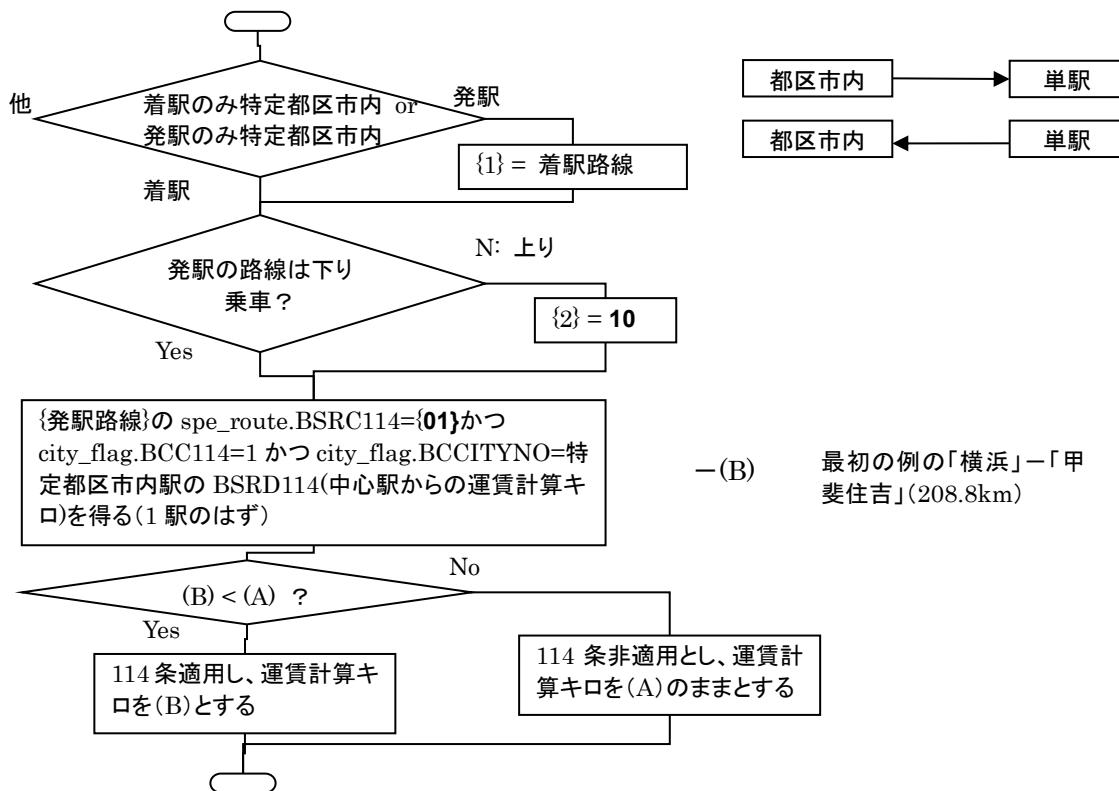
口は 180~200.0km or 80~100km

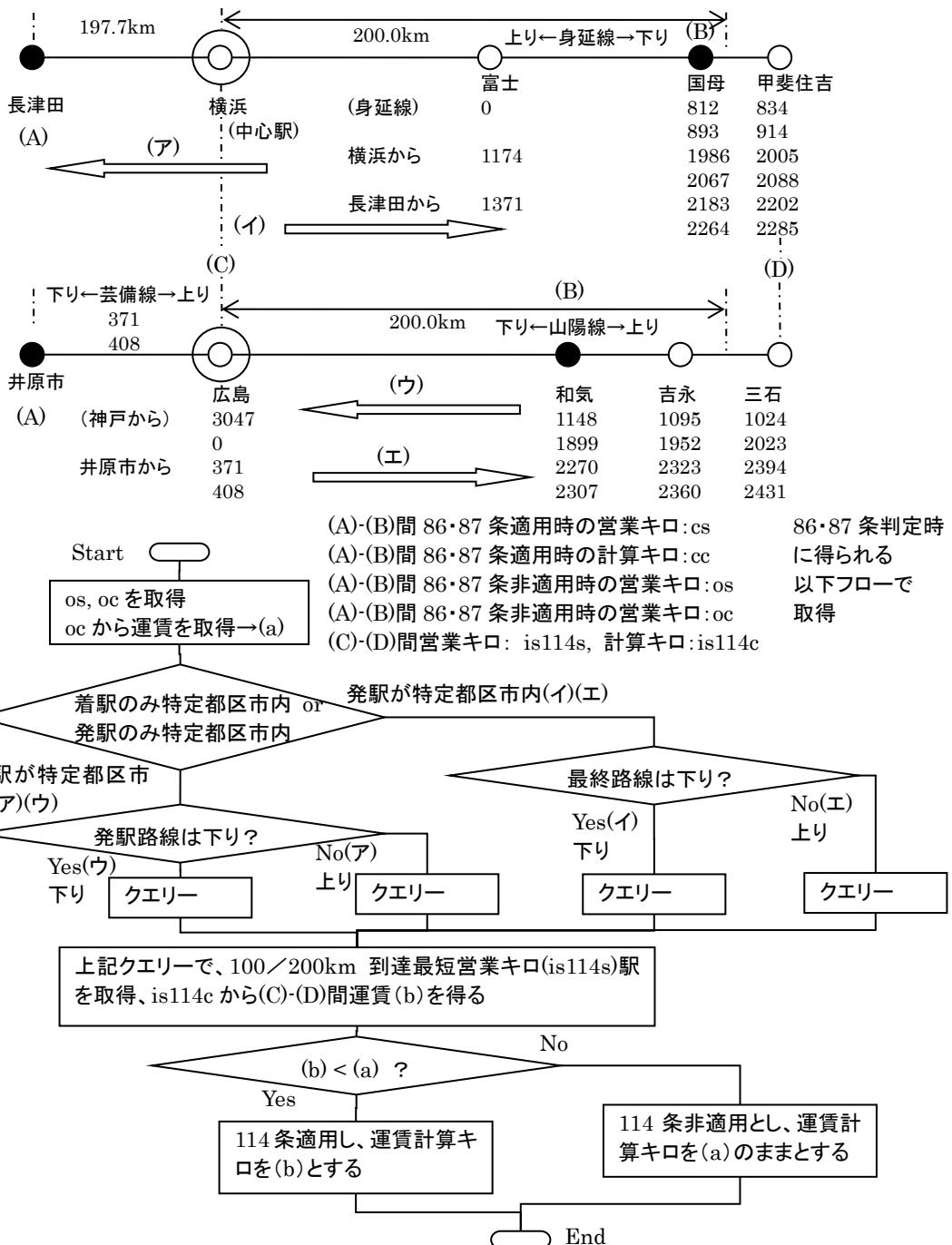
について以下判定処理を実施(それ以外は単駅の運賃計算キロ(下記(A))のままでし 114 条非適用とする)

#### 判定処理

- 単駅の運賃計算キロを算出 - (A)

(最初の例の「長津田」-「国母」間の運賃計算キロ 226.4km)





井原市－和氣 227.0km／¥3,570

広島－和氣 189.9km／¥3,260

### クエリー

\*\*\*広島(3143)から下り方向で200km 到達地点

```

select 11.sales_km-12.sales_km as sales_km, 11.station_id
from t_lines 11 join t_lines 12
where 11.line_id=139 and 12.line_id=139
and 11.sales_km>12.sales_km
and 12.station_id=3143
and (11.sales_km-12.sales_km)>2000 order by 11.sales_km limit(1);
  
```

\*\*\*\*広島(3143)から上り方向で 200km 到達地点

```
select l2.sales_km-l1.sales_km as sales_km, l1.station_id
from t_lines l1 join t_lines l2
where l1.line_id=139 and l2.line_id=139
and l2.sales_km>l1.sales_km
and l2.station_id=3143
and (l2.sales_km-l1.sales_km)>2000 order by l1.sales_km desc limit(1);
```

## ■ 規則 157 条 2 項

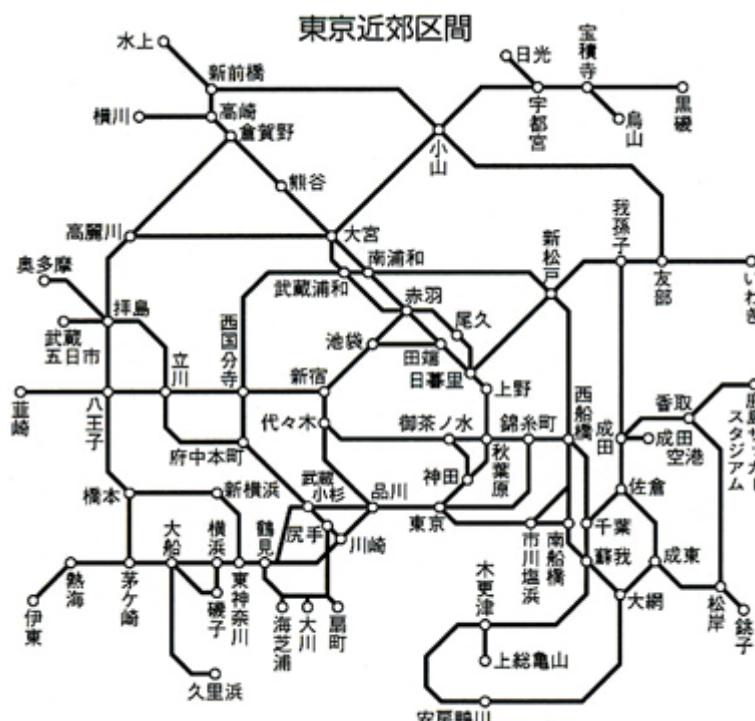
第 157 条 2 項で定める大都市近郊区間内発着の場合、最短経路で計算し、有効日数は 1 日となる(ただし、新幹線利用時はこのルールは適用されない)

発駅または着駅が下記近郊区間外であっても、70 条適用駅を通過した場合は、本規則が適用される。この場合、70 条適用駅を 2 回通過した場合無効で、1 回通過した場合は、70 条適用駅の外端駅同士の最短距離を得て計算する。

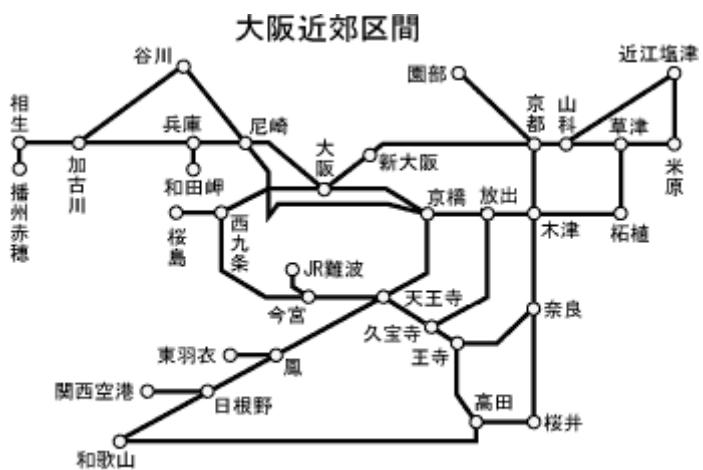
途中下車を望む場合、新幹線を経由に含め非適用にする。この場合の有効日数は乗車経路の営業キロとする。

### データ定義

テーブル t_station		
BCSUBURB	近郊区間	4 ビット
	1 東京	
	2 新潟	
	3 大阪	
	4 福岡	

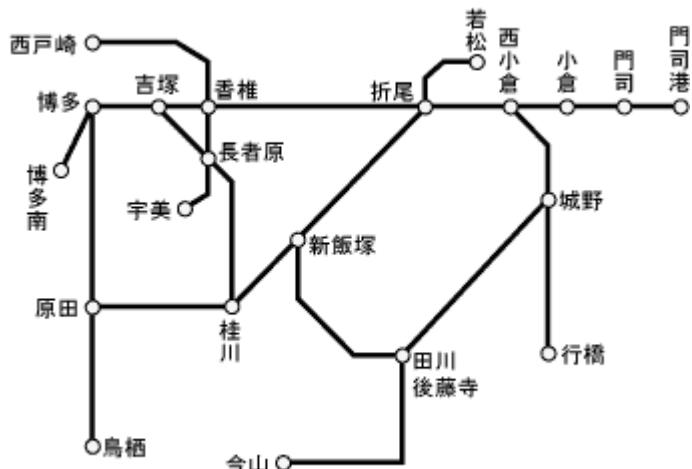


※新幹線で東京～熱海間、東京～那須塩原間、東京～高崎間をご利用になる場合は含まれません。



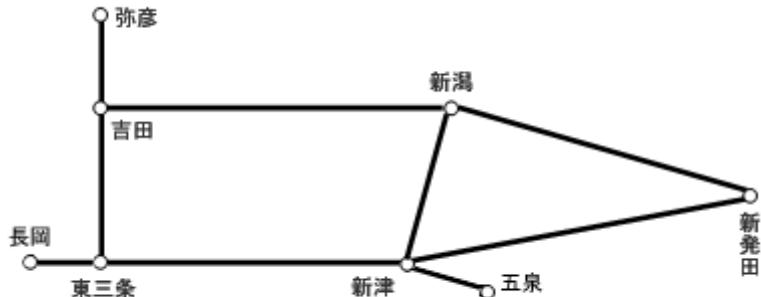
※新幹線で新大阪～西明石間をご利用になる場合は含まれません。

### 福岡近郊区間



※新幹線で小倉～博多間をご利用になる場合は含まれません。

### 新潟近郊区間



※新幹線で長岡～新潟間をご利用になる場合は含まれません。

## 実装詳細

経路がすべて同一近郊区間内である場合、適用される(新幹線など除外条件あり)。

( **最短** ボタンが有効になる)

運賃計算時にチェックされる(一番最初におこなう)。

発駅～着駅までの最短距離をダイクストラ法で算出する。

伊東一水上

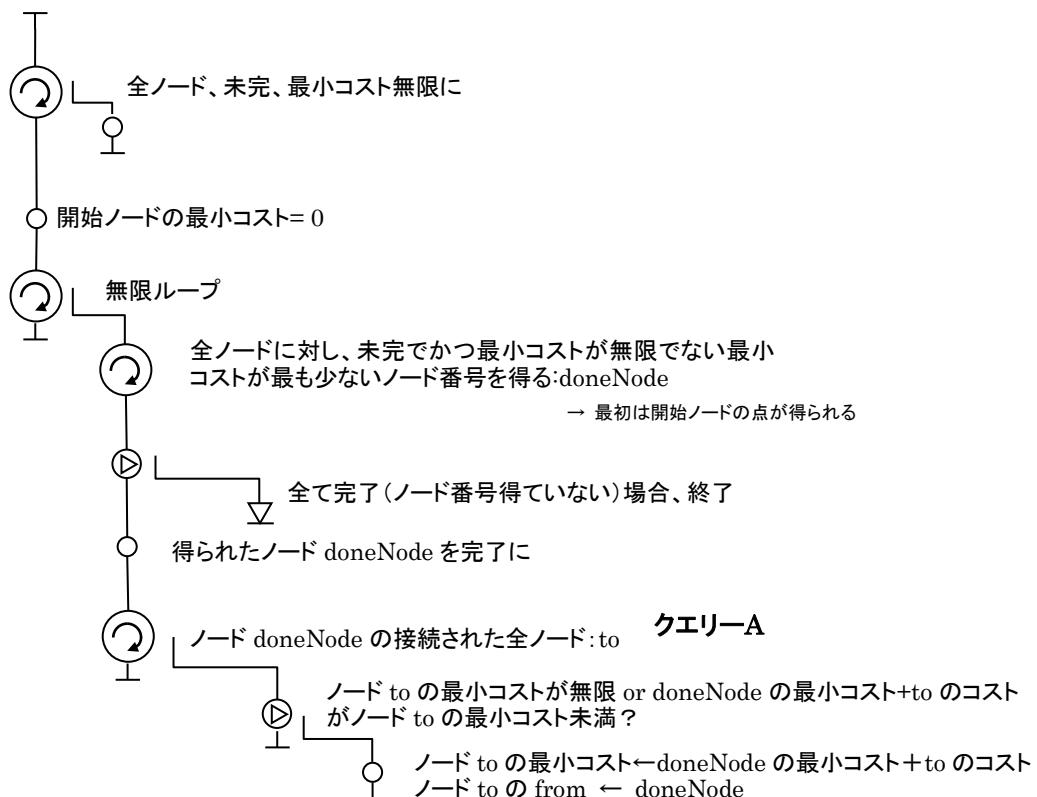
東海道、相模、横浜、八高	260.5km( 269.7km)	¥4620
東海道、相模、横浜、八高、高麗川、川越、大宮	300.5km(303.6km)	¥5250
東海道、相模、横浜、中央東、武蔵野、埼京、大宮	286.7km	¥4940
東海道、相模、横浜、中央東、山手、埼京、大宮	303.3km	¥5250
東海道、東北	285.6km	¥4940

※ DFS、SWA-MARS は 4940、乗換案内は 4620

ノード作成は分岐駅のみ

中間駅は全方向の分岐駅までのコストを得る。

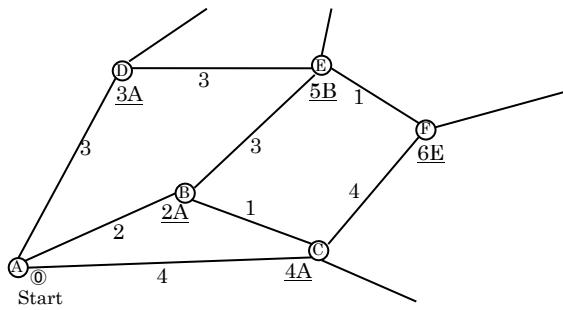
得られた、全分岐駅からの最短経路を算出して決定



jroute.py で実装 SQL の最適化、特別なデータ作成することなく上記フローを実行したところ、i5 2.3GHz で 1 秒強  
新幹線対策未

同一計算キロの場合、どちらかひとつのルートが選択されるがほかの候補の選択は不明である。

同一運賃でも計算キロ(営業キロ)の少ない方が選択され、速度は全く考慮していないため、東京—大阪間は関西本線経由が選択されるし、函館—札幌も函館線経由が選択される。



経路探索は分岐駅のみでおこなわれる。

関数 `dijkstra()`を終了すると Start からのすべての最短距離が全分岐駅にマークされる。

上記の図において、丸囲みのアルファベットは `NameOfNode`、数値は `CostOfSection`、  
`CostOfRoute` であり、`CostOfRoute` と一緒に記載されているアルファベットは `FromNode` をしめします。

関数 `dijkstra(StartOfNode)` 実行後、最終ノード(`EndOfNode`)は任意のノードを選択できます。

上図において、F 点を着駅とすると、F 点をリストの先頭へ追加、次に F 点に隣接している全ノードの最短コストノードは E 点であるので、E 点をリストへ追加、同様に E 点は 5B であり、隣接ノードはコスト 5 で B 点なので B 点をリストへ追加、さらに同様に A 点を追加して終了(A 点=発駅=開始点)。リストを逆順に並べ替えると最短経路となる。

分岐駅以外の駅の開始、終了は以下の方法となります。

#### ○ 開始駅が分岐駅以外の時

- ・ 開始駅の隣接する両端の分岐駅 A,B を得る(一つしか得られなかった場合そのひとつのみ対象)、
- ・ A、B の最小コストを開始駅からそれぞれ A または B までの計算キロとして初期化する  
(上図 HCP チャートの「開始ノードの最小コスト= 0」と記述された部分)
- ・ A、B の `FromNode` を開始駅に設定(上図 HCP チャートの「ノード to の `from ← doneNode`」の部分)
- ・ ※ `done(完了)` フラグは ON にはしない
- ・ 以上の初期化後、関数 `dijkstra()` を実行する(上図 HCP チャート)

#### ○ 終了駅が分岐駅以外の時

- ・ `dijkstra()` を実行
- ・ 終了駅の隣接する両端の分岐駅 A、B を得る(一つしか得られなかった場合、その駅を終了駅とする)
- ・ 終了駅からそれぞれ A または B までの計算キロを `Ae`、`Bc` とする

- ・ (A の最小コスト+Ac) < (B の最小コスト+Bc)なら、終了駅を A、Else なら B となります(最小コストは dijkstra() 実行後に得られている)

dijkstra(開始駅)実行後、終了駅から fromNode を開始駅まで辿れば、最短ルートが得られる。

駅から路線を得る

```
select line_id from t_lines where station_id=?
```

路線、駅から隣接する両端の分岐駅とそこまでの計算キロを得る(?1:路線／?2:駅)(クエリーA)

```
select station_id, (select case when calc_km>0 then calc_km else sales_km end
from t_lines where line_id=?1 and station_id=?2)-case when calc_km>0 then calc_km
else sales_km end as cost
from t_lines
where line_id=?1
and sales_km=(select max(sales_km)
               from t_lines l
               join t_station t
               on t.rowid=l.station_id
               where line_id=?1
               and jctflg<>0
               and sales_km<(select sales_km
                               from t_station t, t_lines l
                               where t.rowid=l.station_id
                               and jctflg<>0
                               and line_id=?1
                               and station_id=?2))
union
select station_id, case when calc_km>0 then calc_km else sales_km end - (select
case when calc_km>0 then calc_km else sales_km end from t_lines where line_id=?1
and station_id=?2) as cost
from t_lines
where line_id=?1
and sales_km=(select min(sales_km)
               from t_lines l
               join t_station t
               on t.rowid=l.station_id
               where line_id=?1
               and jctflg<>0
               and sales_km>(select sales_km
                               from t_station t, t_lines l
                               where t.rowid=l.station_id
                               and jctflg<>0
                               and line_id=?1
                               and station_id=?2))
```

路線、駅から隣接する両端の分岐駅とそこまでの計算キロを得る(?1:路線／?2:駅)(クエリーA)

若干高速版

```
select id, (select case when calc_km>0 then calc_km else sales_km end from t_lines
where line_id=?1 and station_id=?2)-case when calc_km>0 then calc_km else sales_km
end as cost
from t_lines l join t_jct j on j.station_id=l.station_id
where line_id=?1
and sales_km=(select max(sales_km)
               from t_lines l
               join t_jct j
               on j.station_id=l.station_id
               where line_id=?1
               and sales_km<(select sales_km
```

```

        from      t_lines
        where    line_id=?1
                  and    station_id=?2))

union
select  id, case when calc_km>0 then calc_km else sales_km end - (select case when
calc_km>0 then calc_km else sales_km end from t_lines where line_id=?1 and
station_id=?2) as cost
from    t_lines l join t_jct j on j.station_id=l.station_id
where   line_id=?1
and sales_km=(select min(sales_km)
               from t_lines l
               join t_jct j
               on j.station_id=l.station_id
               where line_id=?1
               and sales_km>(select sales_km
                               from t_lines
                               where line_id=?1
                               and station_id=?2))

```

HCP チャートのクエリーA の実行がダイクストラの速度のボトルネックとなる。

分岐駅(t\_jct のレコード)の数は 310 件程度で大したことではないので最適化しなくても処理速度はかからないが、上記のクエリーA は 970 回実行される部分で処理速度の大半を占める。

この結果をDBに保持しておくことにより(t\_jct テーブルに next\_id, cost 列を追加)処理の速度の高速化が図れる。

python 実装の以下の部分(疑似コード)の結果をテーブル作成

```

for jct_id in (select id from t_jct):
    jct_id, cost, line_id in node_next(jct_id)

```

クエリーA の部分は、以下クエリーで得る(python コードの node\_item に相当)

```
-- jct_id と接続する隣の分岐駅リストを得る
select neer_id, cost, line_id from t_node where jct_id=?
```

指定駅が分岐駅か

```
select jctflg from t_station where rowid=?1
```

隣接する両端の分岐駅と計算キロを得る

?1 駅

```

select  station_id , abs((select case when calc_km>0 then calc_km else sales_km
end from t_lines where line_id=(select line_id from t_lines where station_id=?1)
and station_id=?1)-case when calc_km>0 then calc_km else sales_km end) as cost
from    t_lines
where   line_id=(select line_id from t_lines where station_id=?1)
and     sales_km in ((select max(y.sales_km)
                     from t_lines x, t_lines y, t_station t
                     where x.line_id=y.line_id
                     -- and x.line_id=(select line_id from t_lines where
station_id=?1)
                     and x.station_id=?1

```

```
        and t.rowid=y.station_id
        and t.jctflg<>0
        and x.sales_km>y.sales_km) ,
(select min(y.sales_km)
 from t_lines x, t_lines y, t_station t
 where x.line_id=y.line_id
 -- and x.line_id=(select line_id from t_lines where
station_id=?1)
        and x.station_id=?1
        and t.rowid=y.station_id
        and t.jctflg<>0
        and x.sales_km<y.sales_km)) ;
```

## ■ 規則 88 条

大阪・新大阪－姫路以遠は、大阪から計算する。

また、86 条適用時も同様に大阪から計算するように処理する。

86 条チェック&変換の前と後に行う。

	新大阪
東海道線	(神戸)
山陽線	XXXX
:	:

⇒

	大阪
東海道線	(神戸)
山陽線	XXXX
:	:

:	:
:	XXXX
山陽線	(神戸)
東海道線	新大阪

⇒

:	:
:	XXXX
山陽線	(神戸)
東海道線	大阪

XXXX は、山陽線、神戸から姫路までの距離以上にあること

新大阪、東海道線、山陽線をチェックすれば、(神戸)は神戸でしかありえない。

太字部分を置き換え

	大阪
(東海道線)	新大阪
山陽新幹線	YYYY
:	:

⇒

	大阪
東海道線	<b>神戸</b>
<b>山陽線</b>	<b>西明石</b>
山陽新幹線	YYYY
:	:

経由表示しないフラグを追加  
経由表示しないフラグを追加

:	:
:	YYYY
山陽新幹線	新大阪
(東海道線)	大阪

⇒

:	:
:	YYYY
山陽新幹線	<b>西明石</b>
<b>山陽線</b>	<b>神戸</b>
東海道線	大阪

駅のみ経由表示しないフラグを追加  
経由表示しないフラグを追加  
線のみ経由非表示フラグを追加

大阪、新大阪、山陽新幹線をチェックすれば、(東海道線)は自明

YYYY は山陽新幹線に新大阪からの距離が姫路以遠であること

太字部分を置き換え・挿入&フラグ設定

	新大阪
山陽新幹線	YYYY
:	:
:	:

⇒

	大阪
東海道線	<b>神戸</b>
<b>山陽線</b>	<b>西明石</b>
山陽新幹線	YYYY
:	:

経由表示しないフラグを追加  
経由表示しないフラグを追加

:	:
:	YYYY
山陽新幹線	新大阪

⇒

:	:
:	YYYY
山陽新幹線	<b>西明石</b>
<b>山陽線</b>	<b>神戸</b>
東海道線	大阪

駅のみ経由表示しないフラグを追加  
経由表示しないフラグを追加  
路線のみ経由表示しないフラグを追加

## ■ その他規則

規則 89 条

北新地から尼崎以遠へは大阪から計算する(料金のみ。有効期間は通常通り)

→ 個別判定し、自動チェックして計算

九州内山陽新幹線

※ 臨時駅(臨)という接頭辞の駅は、その先までの駅の営業キロで計算するルールの適用を考慮すること

## ■ 会社線通過連絡運輸

会社線内駅で始発、終着することはない(できない)ので、分岐駅のみの運賃表を定義する。

テーブル `t_clinfar` に、各路線につき、 $N = \text{路線内の分岐駅数}$  とすると、 $N \times (N-1)/2$  とおりの運賃表レコードを定義する(以下例)。

青森	野辺地	1010
青森	八戸	2220
青森	二戸	
青森	いわて沼宮内	
青森	好摩	
青森	盛岡	
野辺地	八戸	
野辺地	二戸	
野辺地	いわて沼宮内	
野辺地	好摩	
野辺地	盛岡	
八戸	二戸	
八戸	いわて沼宮内	
八戸	好摩	
八戸	盛岡	
二戸	いわて沼宮内	
二戸	好摩	
二戸	盛岡	
いわて沼宮内	好摩	
いわて沼宮内	盛岡	
好摩	盛岡	

IGR いわて銀河、青い森鉄道は 2 線合わせて「IGR いわて銀河・青い森鉄道」と扱う  
本当は、盛岡—二戸—目時: IGR いわて銀河、目時—八戸—青森: 青い森鉄道

通過するのに、

	新青森
東北線	青森
青い森鉄道	目時
IGR いわて銀河	盛岡
東北線	北上

としなくてよく、

	新青森
東北線	青森
IGR いわて銀河・青い森鉄道	盛岡
東北線	北上

```
select dfare from t_clinfar where cline_id=?1 and ((station_id1=?1 and station_id2=?2) or (station_id1=?2 and station_id2=?1))
```

## ■ 運賃計算

### データ定義

テーブル t\_line

spe\_flg

BSRBORDER

境界駅

1ビット

2駅間の距離

旧

```
select sales_km, calc_km, spe_route from t_lines where line_id=? and (station_id=? or station_id=?)
```

新

```
select l1.sales_km - l2.sales_km, l1.calc_km - l2.calc_km  
from t_lines l1  
join t_lines l2  
where l1.line_id=?1  
and l2.line_id=?1 -- 無くてもいい  
and l1.sales_km>l2.sales_km  
and ((l1.station_id=?2 and l2.station_id=?3)  
or (l1.station_id=?3 and l2.station_id=?2))  
※ 計算キロの場合1行目を以下に変更  
select case when l1.calc_km<>l2_calc_km then l1.sales_km - l2.sales_km else  
l1.calc_km - l2.calc_km end
```

運賃計算はDBではなく、計算により算出する。

計算により算出が難しい、上限が定義できる範囲はDBテーブルに定義する。

テーブルのフィールドは、下限[km]と料金[fare]の2フィールドで以下のようなクエリーで得る

```
select fare_X from t_fareX where km<=? order by km desc limit(1);
```

11, 16, 21, 30

11~15.9, 16~20.9, 21~20.9, 30~

- km は 0 から定義すること
- km の最後は最後に定義された下限値の上限値+1 を定義し、fare は、0 とする（これ以上は定義なしを意味する）

運賃計算は経路配列より以下の乗車キロ累積を計算する

路線	駅	営業キロ	計算キロ	会社種別
	あいの里公園			

札沼線	桑園			
函館線	白石			
千歳線	沼ノ端			



	営業キロ	計算キロ
会社線を除くすべて		
JR 北海道部分		
JR 四国部分		
JR 九州部分		

をそれぞれ算出

#### 本州 3 社と三島会社の境界駅の定義

会社	路線	境界駅	路線	会社
JR 北海道	津軽線	中小国	津軽海峡線	JR 東日本 JR 西日本
JR 四国	瀬戸大橋線	児島	瀬戸大橋線	
JR 九州	山陽線	下関	山陽線	
	鹿児島・日豊線	小倉	山陽新幹線	
	鹿児島・九州新幹線	博多	山陽新幹線	

上記で分岐駅以外の境界駅は、

- 児島(本四備讃線・茶屋町—宇多津間)
- 下関(山陽線・幡生—門司間)

#### 四国

	岡山	JR 西日本
宇野線	茶屋町	JR 西日本
本四備讃線	宇多津	JR 四国
予讃線	高松	JR 四国

#### 九州

	博多	JR 九州
鹿児島線	門司	JR 九州
山陽線	櫛ヶ浜	JR 西日本
岩徳線	岩国	JR 西日本

#### 九州(新幹線-博多)

	熊本	JR 九州
九州新幹線	博多	JR 九州
山陽新幹線	新大阪	JR 西日本

#### 九州(新幹線-小倉)

	新大阪	JR 西日本
山陽新幹線	小倉	JR 九州
鹿児島線	西小倉	JR 九州
日豊線	大分	JR 九州

分岐駅≠境界駅の場合、三島—本州会社のまたがる路線の駅間にある境界駅を得る。

cityflg.BCOSMSP(大阪電車特定区間)、cityflg.BCTKMSP(東京電車特定区間)

### 運賃計算種別

記号	項目	時刻表 JTB/JR	適用距離	キロ 種別	算出方法*1
a	本州三社 - 幹線	A-1/A	営業キロ + 運賃計算キロ	(I)	処理内ハードコーディング fare_a()
b	東京電車区間	D	営業キロ		処理内ハードコーディング fare_b()
c	大阪電車区間	D	営業キロ		処理内ハードコーディング fare_c()
d	山手線内	E	営業キロ		処理内ハードコーディング fare_d()
e	大阪環状線内	E	営業キロ		処理内ハードコーディング fare_e()
f	JR 北 - 幹線	C-1/E	運賃計算キロ		テーブル : t_farem.f, g, h
g	JR 四国	C-1/G	運賃計算キロ (営業キロ + 摘制キロ)	(II)	上限超えは演算 : fare_f0, fare_g0, fare_h
h	JR 九州	C-1/G			
i	本州三社 - 地方交通線	B-1/B	営業キロ	(III)	テーブル t_farel
j	JR 北 - 地方交通線	B-1/F	営業キロ		
k	JR 四国 - 地方交通線のみの特例	C-2/H	摘制キロ 営業キロ	(IV)	テーブル t_farels
l	JR 九州 - 地方交通線のみの特例	C-2/H	摘制キロ 営業キロ		
m	JR 四国の幹線+地方交通線の特例 営業キロ 10km 運賃計算キロ 11km 以内	C-3/I	運賃計算キロ 営業キロ	(V)	処理内ハードコーディング ⇒不要(k, l)に対応 で適用されるので
n	JR 九州の幹線+地方交通線の特例 営業キロ 10km 運賃計算キロ 11km 以内	C-3/I	運賃計算キロ 営業キロ		
o	JR 東+JR 北海道の加算額	A-2/C	運賃計算キロ	(VI)	t_fareopq
p	JR 西+JR 四国の加算額	A-2/C	運賃計算キロ		
q	JR 西+JR 九州の加算額	A-2/C	運賃計算キロ		
r	JR 東+JR 北海道(江差線・海峡線のみ)	B-2/D	営業キロ	t_farer	
s	南千歳・新千歳空港	加算運賃	140	—	t_farest
	日根野・関西空港	加算運賃	210	—	
	日根野・りんくうタウン	加算運賃	150	—	
	りんくうタウン・関西空港	加算運賃	160	—	
	児島・宇多津	加算運賃	100	—	
	田吉・宮崎空港	加算運賃	120	—	
t	その他都市部私鉄競合区間	特定運賃	—	—	
u	会社線	「■ 会社線通過連絡運輸」参照			t_clinfar

\*1 算出関数は、int fare\_a() 関数とする

## 運賃計算ロジック

if 本州 3 社あり?		
if 地方交通線のみ?		
(i) <s>	地方交通線のみ	
else	幹線のみ、幹線+地方交通線	
if 幹線のみ電車特定区間?		
return;		
else		
(a) <s><c> 幹線のみ、幹線+地方交通線 ※「幹線+地方交通線」で全体の営業キロが 10km 以下は (i) <s> で計算		
if JR 北海道あり?		
if JR 北海道側は地方交通線のみ?		
JR 北海道側 (r) <s> 加算		
else	幹線のみ、幹線+地方交通線	
JR 北海道側 (o) <s><c> 加算 ※「幹線+地方交通線」で全体の営業キロが 10km 以下は (r) <s> で計算		
if JR 九州あり?		
JR 九州側 (q) <s><c> 加算	幹線のみ、幹線+地方交通線	
if JR 四国あり?		
四国側 (p) <s><c> 加算	幹線のみ、幹線+地方交通線	
else if JR 北海道		
if 地方交通線のみ	JR 北海道のみ(JR 九州、四国ありはあり得ない)	
(j) <s>		
else	幹線のみ、幹線+地方交通線	
(f) <c>		
else if JR 九州		
(h) <s><c>		JR 九州のみ (JR 北海道、 JR 四国ありは あり得ない)
if <s><c> が 200 km 以下で地方交通線のみ?		
(l) <s><c>	地方交通線のみ	
else		
(n) <s><c> ※10km 以下適用チェック *1	幹線のみ、幹線+地方交通線	
else if 四国		
(g) <s><c>		JR 四国(JR 北 海道、JR 九州 ありはあり得な い)
if <s><c> が 200 km 以下で地方交通線のみ?		
(k) <s><c>	地方交通線のみ	
else		
(m) <s><c> ※10km 以下適用チェック *1	幹線のみ、幹線+地方交通線	
else	空 あり得ない	

( ) 内…運賃計算表

<s> … 営業キロ      <c> … 計算キロ

\*1 地方交通線のみの場合は擬制キロと営業キロ、地方交通線+幹線の場合は運賃計算キロと営業キロ

## ■ 有効日数計算

有効日数は以下の式で算出

```
#!python3.0.1
# -*- coding: utf-8 -*-
# 有効日数

# 営業キロ(×10して小数点第1位を整数)から有効日数を得る

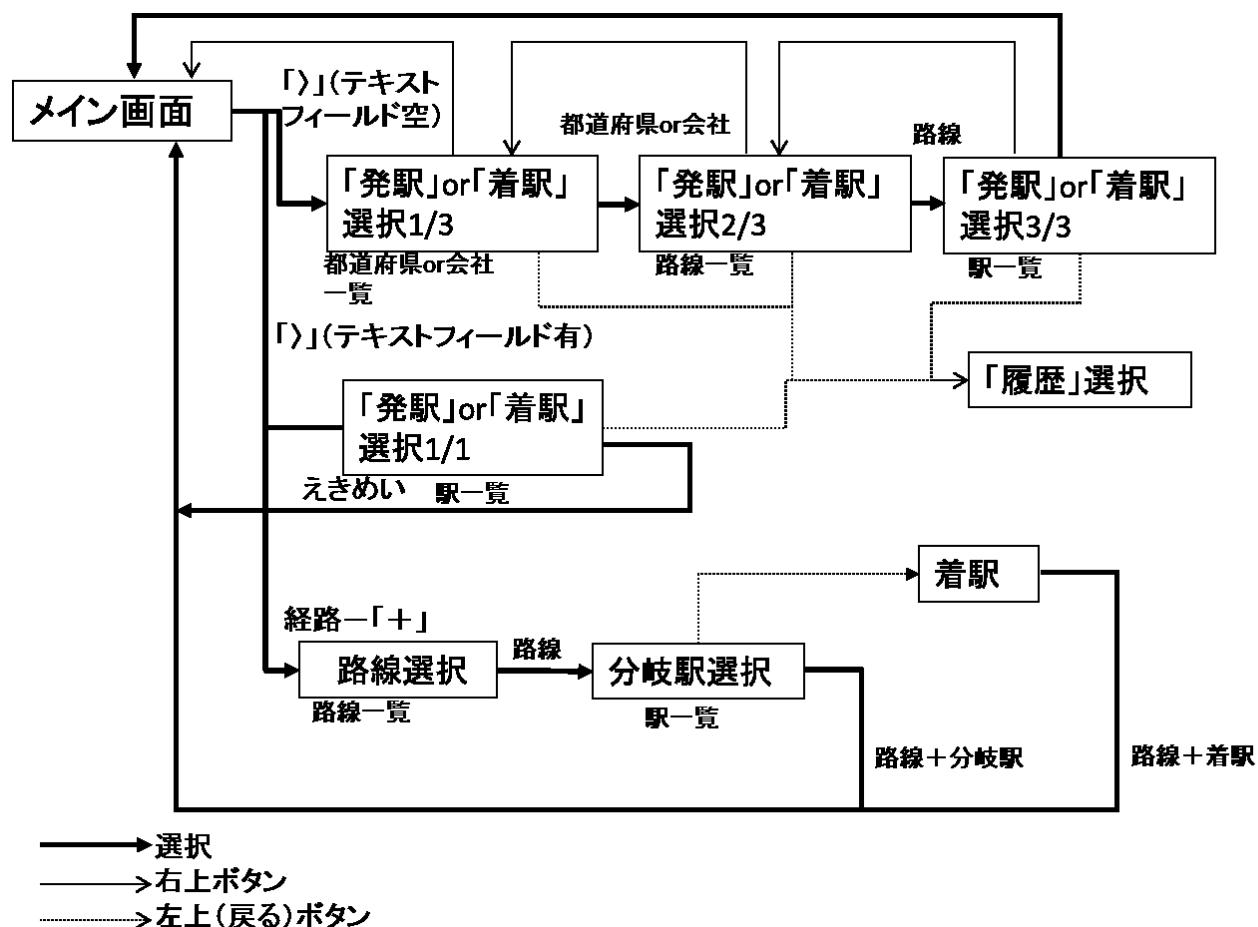
def days_ticket(km):
    if km < 1001:                      # 100 km以下は 1 日
        return 1
    d = (km + 1999) // 2000 + 1
    return d
```

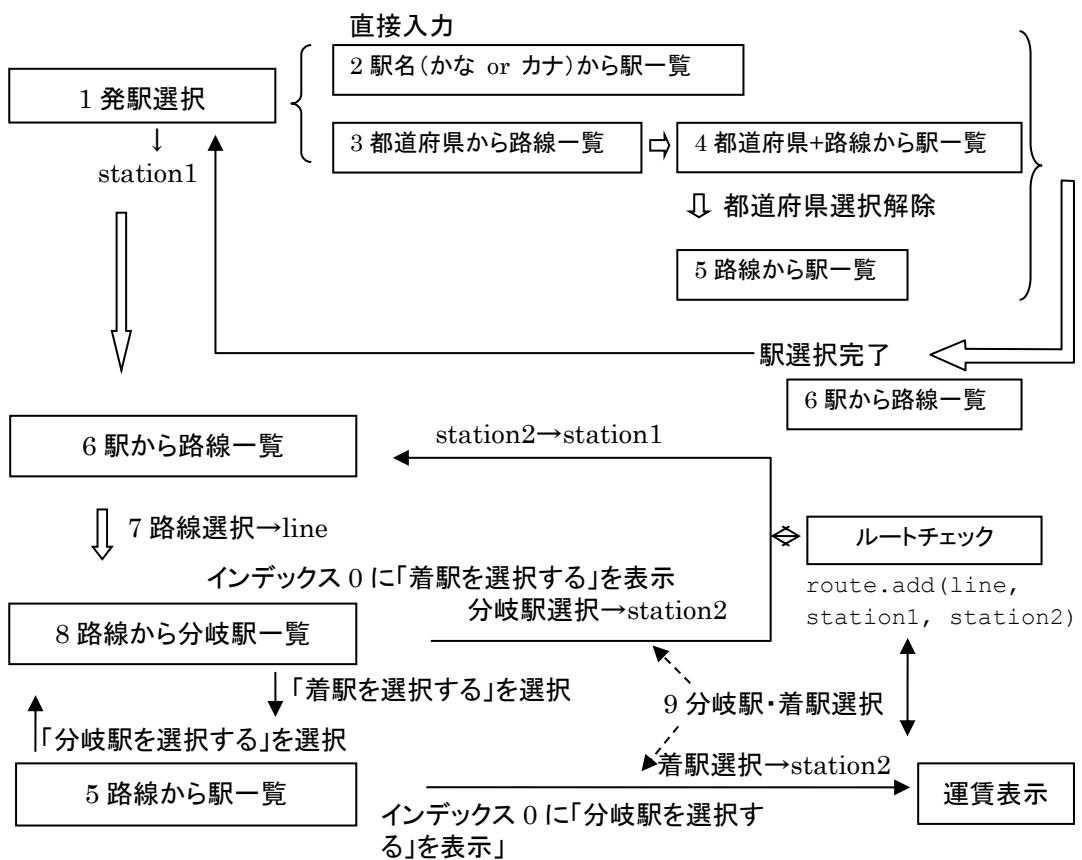
但し、大都市近郊区間内で新幹線を利用しない最短経路運賃計算区間では 1 日とする。

# MVC インタフェース

## ファーストビュー

画面を最初に表示する際には、Route オブジェクトの構築と初期化をおこなう。構築はコンストラクタによるアプリケーション起動時におこない、初期化も同時でも構わない。但し画面が作成されるたびには初期化メソッドとして、Route.clear()は呼び出す必要がある。





## 1.発駅選択

```
route.setStartStation()
```

駅から路線一覧

「駅名から駅名リスト」、「都道府県から路線一覧」、「都道府県+路線から駅一覧」、「路線から駅一覧」、「駅から路線一覧」使用

## 2.駅名から駅名リスト

概要

入力

出力

クエリー

説明(次に想定されるアクションなど)

### 3.都道府県から路線一覧

概要

入力

出力

クエリー

説明(次に想定されるアクションなど)

### 4.都道府県+路線から駅一覧

概要

入力

出力

クエリー

説明(次に想定されるアクションなど)

### 5.路線から駅一覧

概要

入力

出力

クエリー

説明(次に想定されるアクションなど)

Y 字分岐の Ex.1 の例として spe\_route.BSRJCTSP が ON の場合、  
stationId の BSRJCTSP を ON に設定する。

## 6. 駅から路線一覧

概要

入力

出力

クエリー

説明(次に想定されるアクションなど)

Y 字分岐の Ex.2 の例のように「旭川」から「路線一覧」を表示する際、lineId  
の bit30 に spe\_route.BSRJCTSP が ON の情報をコピーする。→  
route.add() を呼ぶときに、lineId.b30 を stationId.b30 へコピーし、  
lineId.b30=0 にして渡す

## 7. 路線選択

概要

route.setStartStation()

路線から分岐駅一覧

## 8. 路線から分岐駅一覧

概要

リスト内のトップインデックスに「全駅(着駅)表示」項目を先頭に。選択すると  
「路線から駅一覧」を表示

入力

出力

クエリー

説明(次に想定されるアクションなど)

## 9.分岐駅／着駅選択

概要

```
route.add()  
route.calcFare()  
「6 駅から路線一覧」表示
```

## 削除選択

Route.removeTail()をコール  
その後、Route.iterator()をコールし、RouteIterator オブジェクトを取得して表示を更新

## クリア選択

Route.clear()をコール

## 運賃計算

運賃計算は常時おこなう

<チェック順>

分岐特例補正<151>

重複チェック

山手線内発着<87>→自動にしない？(最後に適用可否調査後、選択ボタンで適用？)

都区市内発着<86>→自動にしない？(最後に適用可否調査後、選択ボタンで適用？)自動なら 69 の  
後？

特定市内発着料金逆転補正<114>(86 の直後)

新幹線<->在来線乗換  
東京環状線通過<70 条>  
特定選択経路<69 条>  
都市近郊区間最短経路  
↓  
運賃計算ロジック

特別料金  
3 島  
地方交通線のみ  
特定電車区間  
特定私鉄競合区間  
特定区間

# 未整理

2011-7-29

```
select sales_km, calc_km, cityflg, company_id from t_lines l join t_station  
t on t.rowid=l.station_id join t_company c on c.rowid=t.company_id where  
line_id=? and (station_id=? or station_id=?);
```

```
=====
```

```
-----  
query 一覧
```

```
No.summaryinout/*
```

```
1 会社一覧を得るなし会社*
```

```
2 都道府県一覧を得るなし都道府県*
```

```
3 会社から路線会社路線*
```

```
4 都道府県から路線都道府県路線*
```

```
5 漢字から駅駅名-駅*
```

```
6 ひらがなから駅かな駅*
```

```
7 路線から駅路線駅*
```

```
8 路線から分岐駅路線駅*
```

```
9 都道府県+路線から駅都道府県,路線駅 1
```

```
10 会社+路線から駅会社,路線駅 1
```

```
11 駅から路線駅路線*
```

```
12-----
```

```
13 駅から駅の営業キロ、換算キロを得る 駅 1, 駅 2 営業キロ, 換算キロ 1 単一路線
```

```
14 駅名+路線から駅コード駅,路線駅 Id1
```

```
1select name from t_compnay;  
2select name from t_prefect;  
3select distinct(line) from lines where company=? order by line; 'JR 東  
海'  
select n.name, t.company_id from t_lines l join t_station t on  
t.rowid=l.station_id join t_line n on n.rowid=l.line_id join t_company c  
on c.rowid=t.company_id where c.name='JR 東海' group by l.line_id order by  
n.name;  
select n.name, line_id from t_lines l join t_station t on  
t.rowid=l.station_id join t_line n on n.rowid=l.line_id where company_id=3  
group by l.line_id order by n.name;  
4select distinct(line) from lines where prefect=?; '滋賀県'  
select n.name, l.line_id from t_lines l join t_station t on  
t.rowid=l.station_id join t_line n on n.rowid=l.line_id join t_prefect p  
on p.rowid=t.prefect_id where p.name='佐賀県' group by l.line_id;  
select n.name, l.line_id from t_lines l join t_station t on  
t.rowid=l.station_id join t_line n on n.rowid=l.line_id where  
prefect_id=41 group by l.line_id order by n.name;  
5select rowid, name from [t_station] where name like ?:'北千%';  
6select rowid, name from [t_station] where kana like ?:'きた%';  
7select station_id, t.name from t_lines l join t_station t on  
t.rowid=l.station_id join t_line n on n.rowid=l.line_id where n.name='  
小海線' order by sales_km  
8select station_id, t.name from t_lines l join t_station t on  
t.rowid=l.station_id join t_line n on n.rowid=l.line_id where t.jctflg<>0
```

```

and n.name='函館線' order by l.sales_km
9select station_id, t.name from t_lines l join t_station t on
t.rowid=l.station_id join t_line n on n.rowid=l.line_id join t_prefect p
on p.rowid=t.prefect_id where n.name='小海線' and p.name='長野県' order by
sales_km
select station_id, t.name from t_lines l join t_station t on
t.rowid=l.station_id where line_id=90 and prefect_id=5 order by sales_km;
10select station_id, t.name from t_lines l join t_station t on
t.rowid=l.station_id join t_line n on n.rowid=l.line_id join t_company c
on c.rowid=t.company_id where n.name='東海道線' and c.name='JR 東日本' order
by sales_km
select station_id, t.name from t_lines l join t_station t on
t.rowid=l.station_id where line_id=90 and company_id=5 order by sales_km;
11select line from lines where station=?;
select n.name, l.line_id from t_line n join t_lines l on n.rowid=l.line_id
where station_id=508;
12-----
13select sales_km, calc_km from t_lines l join t_station t on
t.rowid=l.station_id join t_line n on n.rowid=l.line_id where (t.name='
追分' or t.name='米沢') and n.name='奥羽線' order by sales_km desc;
14select l.station_id from t_lines l join t_station t on
l.station_id=t.rowid join t_line n on n.rowid=l.line_id where t.name='
大久保' and n.name='中央東線'

```

```

select distinct(line) from lines where company=? order by line; 'JR 東
海'
select line FROM lines S1 WHERE EXISTS ( SELECT * FROM lines S2 WHERE S1.key_1
= S2.key_1 AND S1.key_2 = S2.key_2 AND S1.rowid < S2.rowid)
select line FROM lines S1 WHERE EXISTS ( SELECT * FROM lines S2 WHERE S1.line
= S2.line AND S1.rowid < S2.rowid) where compay='JR 西日本'

```

### 立川-高尾間の分岐駅(結果は、立川と八王子)一覧

```

select * from t_lines l join t_line n on n.rowid=l.line_id join t_station
t on t.rowid=l.station_id where n.name='中央東線' and jctflg!=0 and
(sales_km<=(select sales_km from t_lines l join t_line n on
n.rowid=l.line_id join t_station t on t.rowid=l.station_id where n.name='
中央東線' and t.name='高尾') and
sales_km>=(select sales_km from t_lines l join t_line n on n.rowid=l.line_id
join t_station t on t.rowid=l.station_id where n.name=' 中央東線' and
t.name='立川')) or
(sales_km>=(select sales_km from t_lines l join t_line n on
n.rowid=l.line_id join t_station t on t.rowid=l.station_id where n.name='
中央東線' and t.name='高尾') and
sales_km<=(select sales_km from t_lines l join t_line n on n.rowid=l.line_id
join t_station t on t.rowid=l.station_id where n.name=' 中央東線' and
t.name='立川')) )
↓コードのみ(中央東線 line_id=23, 立川=532、高尾=537

```

```

select id from t_lines l join t_station t on t.rowid=l.station_id join t_jct
j on j.station_id=l.station_id where line_id=23 and jctflg!=0 and
((sales_km<=(select sales_km from t_lines where line_id=23 and
station_id=537) and
sales_km>=(select sales_km from t_lines where line_id=23 and
station_id=532)) or

```

```

(sales_km>=(select sales_km from t_lines where line_id=23 and
station_id=537) and
sales_km<=(select sales_km from t_lines where line_id=23 and
station_id=532)))
↓もっと実用的に(bind(1, line), bind(2, eki1), bind(3, eki2))

```

```

"select t.name from t_lines l join t_line n on n.rowid=l.line_id join
t_station t on t.rowid=l.station_id where n.name=?1 and jctflg!=0 and "
"((sales_km<=(select sales_km from t_lines l join t_line n on
n.rowid=l.line_id join t_station t on t.rowid=l.station_id where n.name=?1
and t.name=?2) and "
"sales_km>=(select sales_km from t_lines l join t_line n on
n.rowid=l.line_id join t_station t on t.rowid=l.station_id where n.name=?1
and t.name=?3)) or "
"(sales_km>=(select sales_km from t_lines l join t_line n on
n.rowid=l.line_id join t_station t on t.rowid=l.station_id where n.name=?1
and t.name=?2) and "
"sales_km<=(select sales_km from t_lines l join t_line n on
n.rowid=l.line_id join t_station t on t.rowid=l.station_id where n.name=?1
and t.name=?3))"

```

```

select id,t.name from t_lines l join t_station t on t.rowid=l.station_id
join t_jct j on j.station_id=l.station_id where line_id=93 and jctflg!=0
and
((sales_km<=(select sales_km from t_lines where line_id=93 and
station_id=1996) and
sales_km>=(select sales_km from t_lines where line_id=93 and
station_id=2050)) or
(sales_km>=(select sales_km from t_lines where line_id=93 and
station_id=1996) and
sales_km<=(select sales_km from t_lines where line_id=93 and
station_id=2050)))

```

```

const char* cmd_sql_linedup =
"select station_id from t_lines where line_id=?1 and (
"station_id=?2 OR "
"station_id=?3 OR "
"station_id=?4 OR "
"station_id=?5) order by sales_km";

```

----

DB をプログラムからどのようにして読み出すか  
Browfish 暗号する  
リソースファイルの保持  
その形式は？

sqlite3

.backup file

```
db file create  
sqlite3_backup_init() api used.  
sqlite3_backup_xxx() api used.
```

```
.dump  
sql statement エコーバック
```

```
.restore file  
.backup file 's restore.と同様以下 API で実現  
sqlite3_backup_init() api used.  
sqlite3_backup_xxx() api used.
```

### 結論

- 1- RC には、Dump して生成された全 SQL 文を Browfish 暗号格納
- 2- RC リード
- 3- RC 復号
- 4- :memory: DB へ Create
- 5- DB へ RC 復号 SQL 文を実行

1. Dump ツールは ?RC への登録 ちょっとめんどいけど頻度は少ないはずだし手間もどうってことないはず。

1. 3. 鍵は ? CBC 初期ベクトル const 変数ハードコード UUIDGEN、  
UUIDGEN の下 4 ビットのスクランブルの暗号のスクランブル。
5. API は ? sql3\_exec() で一括

Temp ファイルを作成が簡単なのだが、拾われる恐れがある。

-> Temp ファイル作成にする。

作成して :memory: open -> sqlite3\_backup\_init() で restore して temp を削除

~~~~~  
<--- 決定

### 理想

memory をファイルとして扱えないか?  
メモリイメージを SQLITE 側でアタッチできないか?

uuidgen の 256 × 2 バイト

## Appendix.

# DB 定義

## テーブル

### t\_company 会社線テーブル

| テーブル名<br>(説明) |      | t_company<br>(会社線テーブル) |    |     |
|---------------|------|------------------------|----|-----|
| Key           | 列    | 型                      | 制約 | 説明  |
| *             | name | char(11)               | pk | 会社名 |

### t\_prefect 都道府県テーブル

| テーブル名<br>(説明) |      | t_prefect<br>(都道府県テーブル) |    |       |
|---------------|------|-------------------------|----|-------|
| Key           | 列    | 型                       | 制約 | 説明    |
| *             | name | char(12)                | pk | 都道府県名 |

### t\_line 路線名テーブル

| テーブル名<br>(説明) |      | t_line<br>(路線名テーブル) |    |                             |
|---------------|------|---------------------|----|-----------------------------|
| Key           | 列    | 型                   | 制約 | 説明                          |
| *             | name | text                |    | 路線名(駅名と重ならない名称は末尾の「線」を省略する) |

※ 新幹線はテーブルの最終レコードに集められ、新幹線か在来線か否かについては最初の新幹線

の id を固定なハードコード定義(#define または final int など)で指定し、その値以上は新幹線、未満は在来線と判断されるようにする。

※ 上越新幹線、北陸長野新幹線は東京から定義され、東京一大宮は、東北新幹線と重複し、大宮一高崎間は、上越新幹線、北陸長野新幹線は重複する

### t\_station 駅テーブル

| テーブル名<br>(説明) |            | t_station<br>(駅テーブル) |                  |                      |
|---------------|------------|----------------------|------------------|----------------------|
| Key           | 列          | 型                    | 制約               | 説明                   |
| *             | name       | text                 |                  | 駅名                   |
|               | kana       | text                 |                  | 駅名読み(かな)             |
|               | company_id | int                  | t_company(rowid) | 所属会社                 |
| *             | prefect_id | int                  | t_prefect(rowid) | 都道府県名                |
|               | samepname  | text                 |                  | 同名駅区別文字(同名駅でなければ空文字) |
|               | sflg       | int                  |                  | フラグ                  |

### t\_lines 路線テーブル

| テーブル名<br>(説明) |            | t_lines<br>(路線テーブル) |                  |                                              |
|---------------|------------|---------------------|------------------|----------------------------------------------|
| Key           | 列          | 型                   | 制約               | 説明                                           |
| *             | line_id    | int                 | t_line(rowid)    | 路線<br>※ 新幹線は在来線のあとに定義<br>※ 通貨連絡運輸の会社線は最後半に定義 |
| *             | station_id | int                 | t_station(rowid) | 駅                                            |
|               | sales_km   | int                 |                  | 下り起点駅からの営業キロ<br>(特例ルートによっては異なる情報が格納)         |
|               | calc_km    | int                 |                  | 下り起点駅からの計算キロ<br>(特例ルートによっては異なる情報が格納)         |
|               | lflg       | int                 |                  | 特例ルートフラグ                                     |

sales\_km=-1: 新幹線で新幹線の駅の無い在来線分岐駅(平行在来線の新幹線でない分岐駅。名古屋、沼津とか新富士は含まず、金山、草津とか山科は含む。また国府津、富士、沼津などは異なる線なので含まない。※分岐駅の列挙「乗車経路マーク」時に使用

calc\_km=-1: 会社線通過連絡運輸の路線の場合

lflg.BSRJCTSP=1: 分岐特例の定義で、calc\_km の下位 16 ビット=分岐駅特例の実(戸籍上の)分岐駅(c) / calc1\_km の上位 2 バイト=実路線(b)を表す。

分岐駅特例駅では、sales\_km を路線の開始駅は、-32768km(中央西線では名古屋とか)、終了駅は、+32767.0km(中央西線では松本とか)と定義するようにした。

### t\_jct 分岐テーブル

| テーブル名<br>(説明) |            | t_jct<br>(分岐テーブル) |                  |    |
|---------------|------------|-------------------|------------------|----|
| Key           | 列          | 型                 | 制約               | 説明 |
| *             | id         | int               | autoincrement    |    |
|               | station_id | int               | t_station(rowid) |    |

### t\_node ノード定義テーブル

| テーブル名<br>(説明) |         | t_node<br>(分岐テーブル) For dijkstra() |              |                        |
|---------------|---------|-----------------------------------|--------------|------------------------|
| Key           | 列       | 型                                 | 制約           | 説明                     |
| *             | jct_id  | int                               | t_jct.id     |                        |
| *             | neer_id | int                               | t_jct.id     | jct_id の隣の分岐駅          |
|               | cost    | int                               |              | jct_id ~ neer_id の計算キロ |
| *             | line_id | int                               | t_line.rowid | 路線名                    |

※ jct\_id と neer\_id は入れ替わった場合も含めた組み合わせでユニーク

**t\_clinfar 会社線駅間料金定義テーブル**

| テーブル名<br>(説明) |             | t_clinfar<br>(会社線駅間料金定義テーブル) |               |               |
|---------------|-------------|------------------------------|---------------|---------------|
| Key           | 列           | 型                            | 制約            | 説明            |
| *             | station_id1 | int                          |               | 駅 1           |
| *             | station_id2 | int                          |               | 駅 2           |
|               | fare        | int                          |               | 駅 1～駅 2 の大人料金 |
|               | line_id     | int                          | t_line(rowid) |               |

**t\_rule69 旅客営業規則 69 条適用テーブル**

| テーブル名<br>(説明) |             | t_rule69<br>(規則 69 条適用テーブル) |                  |          |
|---------------|-------------|-----------------------------|------------------|----------|
| Key           | 列           | 型                           | 制約               | 説明       |
| *             | id          | int                         | 1~8              | 経路 ID    |
|               | station_id1 | int                         | t_station(rowid) | 発:       |
|               | station_id2 | int                         | t_station(rowid) | 至:       |
| *             | line_id     | int                         | t_line(rowid)    | 置き換え後の路線 |
|               | ord         | tinyint                     |                  | 表示順(下り)  |

**t\_rule70 旅客営業規則 70 条適用テーブル**

| テーブル名<br>(説明) |             | t_rule70<br>(規則 70 条適用テーブル) |                  |      |
|---------------|-------------|-----------------------------|------------------|------|
| Key           | 列           | 型                           | 制約               | 説明   |
| *             | station_id1 | int                         | t_station(rowid) |      |
| *             | station_id2 | int                         | t_station(rowid) |      |
|               | sales_km    | int                         |                  | 営業キロ |

**t\_rule86 旅客営業規則 86 条、87 条定義テーブル**

| テーブル名<br>(説明) |            | t_rule86<br>旅客営業規則 86 条、87 条定義テーブル |                  |                                      |
|---------------|------------|------------------------------------|------------------|--------------------------------------|
| Key           | 列          | 型                                  | 制約               | 説明                                   |
| *             | line_id1   | int                                | t_line(rowid)    | 脱出(進入)路線                             |
|               | station_id | int                                | t_station(rowid) | 乗換駅                                  |
|               | line_id2   | int                                | t_line(rowid)    | 中心駅発路線                               |
| *             | city_id    | int                                |                  | 都区市内番号(1～11)<br>複数レコードの場合、2 レコード目以降、 |

|  |  |  |  |                     |
|--|--|--|--|---------------------|
|  |  |  |  | +0x100, +0x200, ... |
|--|--|--|--|---------------------|

- ※ line\_id2 の路線は city\_id でしめす都区市内中心駅を有する路線である必要があります
- ※ station\_id は、line\_id1 にも line\_id2 にも接続する駅である必要があります
- ※ 脱出(進入)路線が、都区内中心駅から 2 路線を越える場合、同一の line\_id1 の 2 レコード定義となる(おおさか東線)。この場合、city\_id の bit15-8 までにレコード番号(2 つめ=1 即ち+0x10 した値)が格納される。

### t\_farem 3 島会社幹線キロ程定義テーブル

| テーブル名<br>(説明) |    | t_farem<br>キロ程定義テーブル(幹線、基準) |    |                            |
|---------------|----|-----------------------------|----|----------------------------|
| Key           | 列  | 型                           | 制約 | 説明                         |
| *             | km | int                         |    | 運賃計算キロの下限                  |
|               | f  | int                         |    | JR 北海道のキロ別料金(=0 で上限越えをしめす) |
|               | g  | int                         |    | JR 四国のキロ別料金(=0 で上限越えをしめす)  |
|               | h  | int                         |    | JR 九州のキロ別料金(=0 で上限越えをしめす)  |

km は 0 から定義すること

km の最後は最後に定義された下限値の上限値+1 を定義し、fare は、0 とする(これ以上は定義なしを意味する)

### t\_farel 地方交通線キロ程定義テーブル

| テーブル名<br>(説明) |    | t_farel<br>キロ程定義テーブル(地方交通線のみ利用) |    |                   |
|---------------|----|---------------------------------|----|-------------------|
| Key           | 列  | 型                               | 制約 | 説明                |
| *             | km | int                             |    | 運賃計算キロの下限         |
|               | i  | int                             |    | 本州 3 社の地方交通線キロ別料金 |
|               | j  | int                             |    | JR 北海道の地方交通線キロ別料金 |

### t\_farels 3 島会社地方交通線キロ程定義テーブル特例

| テーブル名<br>(説明) |     | t_farels<br>キロ程定義テーブル |    |           |
|---------------|-----|-----------------------|----|-----------|
| Key           | 列   | 型                     | 制約 | 説明        |
| *             | dkm | int                   |    | 擬制キロ      |
|               | skm | int                   |    | 営業キロ      |
|               | k   | int                   |    | JR 四国適用料金 |
|               | l   | int                   |    | JR 九州適用料金 |

### t\_fareopq 本州 3 社+3 島会社加算

| テーブル名<br>(説明) |    | t_fareopq<br>本州 3 社+3 島通過加算テーブル |    |           |
|---------------|----|---------------------------------|----|-----------|
| Key           | 列  | 型                               | 制約 | 説明        |
| *             | km | int                             |    | 運賃計算キロ    |
|               | o  | int                             |    | JR 北海道加算額 |
|               | p  | int                             |    | JR 四国加算額  |
|               | q  | int                             |    | JR 九州加算額  |

### t\_farer 本州 3 社+JR 北海道地方交通線会社加算

| テーブル名<br>(説明) |    | t_farer<br>JR 東日本+JR 北海道地方交通線通過加算テーブル |    |           |
|---------------|----|---------------------------------------|----|-----------|
| Key           | 列  | 型                                     | 制約 | 説明        |
| *             | km | int                                   |    | 営業キロ      |
|               | r  | int                                   |    | JR 北海道加算額 |

### t\_farest 特定区間加算

| テーブル名<br>(説明) |          | t_fares<br>特定区間通過加算テーブル |    |                                  |
|---------------|----------|-------------------------|----|----------------------------------|
| Key           | 列        | 型                       | 制約 | 説明                               |
| *             | station1 | int                     |    | 発着駅 1 の station_id               |
| *             | station2 | int                     |    | 発着駅 2 の station_id               |
|               | fare     | int                     |    | 料金                               |
|               | kind     | int                     |    | =0: 特定区間通過加算(s)<br>≠0: 特定区間料金(t) |

### t\_hzline 新幹線の並行在来線

| テーブル名<br>(説明) |         | t_hzline<br>新幹線の並行在来線 |               |          |
|---------------|---------|-----------------------|---------------|----------|
| Key           | 列       | 型                     | 制約            | 説明       |
| *             | line_id | int                   | t_line(rowid) | 並行在来線 ID |

line\_id が 65535 の場合、境界駅であることをしめします(上越新幹線の高崎)

このテーブルの行 id は、t\_lines の lflg のビット 16-13(BSRHZLIN)に定義される。

## 旧定義

### spe\_route の整理

| テーブル      |            | 列                                                                                                                                                                                                                                                                       |
|-----------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| t_lines   |            | spe_route                                                                                                                                                                                                                                                               |
| ビット       | ラベル        | 説明                                                                                                                                                                                                                                                                      |
| 31        | BSRJCTSP   | 分岐駅特例、 or <rule1>適用したら ON, しない場合 OFF にする                                                                                                                                                                                                                                |
| 30        |            | 0 (RouteItem 挿入時、発駅側が分岐特例の場合に ON (<rule3>) にする                                                                                                                                                                                                                          |
| 29        | BSR69TERM  | 営業規則 69 条に該当する路線・分岐駅(端駅 1、端駅 2)に定義(中間駅には該当しない)。このビットが ON の場合、bit3-0 は営業規則 69 条の項番も設定される<br>※「営業規則 69 条最短経路で計算」ルート検出に使用                                                                                                                                                  |
| 28        | BSR69CONT  | 営業規則 69 条に該当する路線・分岐駅(端駅 1、端駅 2)に定義(中間駅には該当しない)で、ON の場合は路線にまたがっており、継ぎがあることをしめします(東海道線、東海道新幹線、北陸線の米原に該当)<br>このビットが ON の場合、bit3-0 は営業規則 69 条の項番も設定される<br>※「営業規則 69 条最短経路で計算」ルート検出に使用                                                                                       |
| 27        |            | 新幹線の在来線分岐駅(平行在来線の新幹線がない分岐駅。名古屋、沼津とか新富士は含まず、草津とか山科は含む)<br>※ 分岐駅の列挙「乗車経路マーク」時に使用                                                                                                                                                                                          |
| 26        |            | 新幹線の平行在来線で乗換駅であると ON(在来線側のみで新幹線側は無効)<br>以下の 25-24 ビットとセットで使用される<br>※ 新幹線の並行在来線取得に使用                                                                                                                                                                                     |
| 25-2<br>4 |            | 平行在来線 ⇄ 新幹線乗り換え時の侵入許可フラグ(並行在来線側のみで新幹線側は無効)<br>00=通常(豊橋、浜松、新横浜、東京、新富士など)<br>01=新幹線降車後、新幹線下り線 → 平行在来線上り有効または、在来線から新幹線乗車時、在来线下り線 → 新幹线上り有効(静岡など)<br>10=新幹線降車後、新幹线上り線 → 平行在来线下り有効または、在来線から新幹線乗車時、在来线上り線 → 新幹线下り有効(名古屋、三島など)<br>11=上下線、平行在来线上线下有効(広島など)<br>※「新幹線と在来線乗り換え」で使用 |
|           |            | ON で新幹線駅(平行在来線駅併設)新大阪とか(吉川新花巻は、OFF で jctflg が ON)<br>ON で新幹線の在来線分岐駅で分岐駅特例乗車可能区間の本来の分岐駅(例: 金山)これが ON の場合、ビット 27 も ON である<br>ON で新幹線駅(新幹線のみ。新幹線の沼津などは OFF)<br>未割当                                                                                                         |
| 23        | BSR70      | 規則 70 条定義レコード ON のとき、line_id 列は 70 条用に払い出し station_id は、b0～b15 の路線の計算駅<br>b0～15 は進入 or 脱出路線                                                                                                                                                                             |
| 22        | BSRVIRTUAL | 仮想駅(営業キロ計算用で、リスト表示時、分岐駅マーク時には適用しない駅)                                                                                                                                                                                                                                    |
| 21-2<br>0 | BSRC114    | city_flg.BCC114-1 のときに有効<br>00: 本路線では無効<br>01: 下り線で有効<br>10: 上り線で有効<br>11: 上下線で有効(否定義)                                                                                                                                                                                  |
| 19-1<br>1 | BSRD114    | BSRC114 ≠ 00 のときのみ有効<br>中心駅からの運賃計算キロをしめす。87 条の場合、100km(1000) を引いた値、86 条の場合、200km(2000) を引いた値(0～511)<br>※ 100/200km に達する駅が 151.1/251.1km を越える場合<br>151.1/251.1km に切り捨てて定義する(それでも可のはず)<br>t_station.jctflg の内容をここに含む                                                      |
|           | BSRBORDER  | 境界駅                                                                                                                                                                                                                                                                     |
| 新幹線の場合    |            |                                                                                                                                                                                                                                                                         |
| 8-0       |            | 0～511 並行在来線路線 id                                                                                                                                                                                                                                                        |

|                 |                 |                                    |
|-----------------|-----------------|------------------------------------|
| 在来線の場合          |                 |                                    |
|                 |                 |                                    |
|                 |                 |                                    |
| BSR70=1 の場合     |                 |                                    |
| 14-0            | BSR70MASK       | 大都市環状線 (70 条適用駅) エリアへの進入 or 脱出路線   |
| 在来線の場合(bit29=1) |                 |                                    |
| 3-0             | BSR69NOMA<br>SK | 0~8 営業規則 69 条の項番に相当(経路 ID)t_rule69 |

```
#define FLGTEST(flg, MASK) (flg & MASK)
#define FLGVALU(flg, MASK, POS) ((flg & MASK) >> POS)
// 一般路線
#define BSRNMLMASK (~(BSR70 | BSRVIRTUAL |
```

### city\_flg の整理

| テーブル      | 列        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
|-----------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-------|------|----|------|------|---|-------|------|---|------|------|---|------|------|---|------|------|---|------|------|---|-------|------|---|------|------|----|------|------|----|------|------|
| t_station | cityflg  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| ビット ラベル   | 説明       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 31-21     | 空き       | t_station.jctflg を廃止してこのビットに割り当てる<br>さらに t_lines.spe_route にも含ませる<br>0=分岐駅なし、1=分岐駅あり                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| BCOSMSP   | 大阪電車特定区間 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| BCTKMS    | 東京電車特定区間 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 20-17     | BCSUBURB | 近郊区間 <table border="1" style="margin-left: 20px;"> <tr><td>1</td><td>東京</td></tr> <tr><td>2</td><td>新潟</td></tr> <tr><td>3</td><td>大阪</td></tr> <tr><td>4</td><td>福岡</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                             | 1 | 東京    | 2    | 新潟 | 3    | 大阪   | 4 | 福岡    |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 1         | 東京       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 2         | 新潟       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 3         | 大阪       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 4         | 福岡       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 16        | BCRULE70 | 旅客営業規則 70 条適用駅                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 15        | BCYAMATE | 山手線内駅 <p>都区市内代表駅からの営業キロ×10km (ビット4=1のときのみ有効、ビット4=0のときは無効)<br/>都区市内代表駅一覧は「エラー! 参照元が見つかりません。」参照</p> <p>1のとき、b14-5の10ビットが都区市内センターからの営業キロ (範囲の再外側駅のみ)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                               |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
|           | BCC114   | 都区市内中心駅から 100km、200km に達する最初の駅 (114 条チェック用)<br>(BCCITYNO≠0かつ、spe_route.BSRC114≠0 の路線でのみ有効)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 4         | BCCITYCT | 都区市内中心駅の場合 1(BCCITYNO≠0 であること)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 3-0       | BCCITYNO | 都区市内番号 (1~12) <table border="1" style="margin-left: 20px;"> <tr><td>1</td><td>東京都区内</td><td>0x01</td></tr> <tr><td>2</td><td>横浜市内</td><td>0x02</td></tr> <tr><td>3</td><td>名古屋市内</td><td>0x03</td></tr> <tr><td>4</td><td>京都市内</td><td>0x04</td></tr> <tr><td>5</td><td>大阪市内</td><td>0x05</td></tr> <tr><td>6</td><td>神戸市内</td><td>0x06</td></tr> <tr><td>7</td><td>広島市内</td><td>0x07</td></tr> <tr><td>8</td><td>北九州市内</td><td>0x08</td></tr> <tr><td>9</td><td>福岡市内</td><td>0x09</td></tr> <tr><td>10</td><td>仙台市内</td><td>0x0a</td></tr> <tr><td>11</td><td>札幌市内</td><td>0x0b</td></tr> </table> | 1 | 東京都区内 | 0x01 | 2  | 横浜市内 | 0x02 | 3 | 名古屋市内 | 0x03 | 4 | 京都市内 | 0x04 | 5 | 大阪市内 | 0x05 | 6 | 神戸市内 | 0x06 | 7 | 広島市内 | 0x07 | 8 | 北九州市内 | 0x08 | 9 | 福岡市内 | 0x09 | 10 | 仙台市内 | 0x0a | 11 | 札幌市内 | 0x0b |
| 1         | 東京都区内    | 0x01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 2         | 横浜市内     | 0x02                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 3         | 名古屋市内    | 0x03                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 4         | 京都市内     | 0x04                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 5         | 大阪市内     | 0x05                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 6         | 神戸市内     | 0x06                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 7         | 広島市内     | 0x07                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 8         | 北九州市内    | 0x08                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 9         | 福岡市内     | 0x09                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 10        | 仙台市内     | 0x0a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 11        | 札幌市内     | 0x0b                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |

### spe\_route(lflag) の整理

| テーブル        |               | 列                                                                                                                                                                                                                                                                      |
|-------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| t_lines     |               | spe_route                                                                                                                                                                                                                                                              |
| ビット         | ラベル           | 説明                                                                                                                                                                                                                                                                     |
| 31          | BSRJCTSP      | 分岐駅特例, or <rule1>適用したら ON, しない場合 OFF にする<br>経路追加時には off して追加されるが、ON のときは、removeTail() 時、直前のインデックスも含めて削除(計 2 行)                                                                                                                                                         |
| 30          | BSRJCTFLG     | 0 (RouteItem 揃入時、発駅側が分岐特例の場合に ON (<rule3>) にする<br>(このビットが 1 のときは DB エラーとする)                                                                                                                                                                                            |
| 29          | BSR69TERM     | 営業規則 69 条に該当する路線・分岐駅(端駅 1、端駅 2)に定義(中間駅には該当しない)。このビットが ON の場合、bit BSR69NOMASK は営業規則 69 条の項番も設定される<br>※「営業規則 69 条最短経路で計算」ルート検出に使用                                                                                                                                        |
| 28          | BSR69CONT     | 営業規則 69 条に該当する路線・分岐駅(端駅 1、端駅 2)に定義(中間駅には該当しない)で、ON の場合は路線にまたがっており、継ぎがあることをしめします(東海道線、東海道新幹線、北陸線の米原に該当)<br>このビットが ON の場合、BSR69TERM は必ず ON で、bit BSR69NOMASK は営業規則 69 条の項番も設定される<br>※「営業規則 69 条最短経路で計算」ルート検出に使用                                                          |
| 27          | BSRSHINKTRS   | 新幹線の平行在来線で乗換駅であると ON(在来線側のみで新幹線側は無効)<br>以下の 2 ビットとセットで使用される<br>※ 新幹線の並行在来線取得に使用<br><br>※ 使用しなくても良いらしい<br>※ => 使うことになった(分岐特例にて)でもなくせる<br>(16-13 を使用することで)                                                                                                               |
| 26-25       | BSRSHINKTRSLW | 平行在来線 ⇄ 新幹線乗り換え時の侵入許可フラグ(並行在来線側のみで新幹線側は無効)<br>00=通常(豊橋、浜松、新横浜、東京、新富士など)<br>01=新幹線降車後、新幹線下り線 → 平行在来線上り有効または、在来線から新幹線乗車時、在来線下り線 → 新幹線上り有効(静岡など)<br>10=新幹線降車後、新幹線上り線 → 平行在来线下り有効または、在来線から新幹線乗車時、在来线上り線 → 新幹线下り有効(名古屋、三島など)<br>11=上下線、平行在来線上下有効(広島など)<br>※「新幹線と在来線乗り換え」で使用 |
| 24          | BSR70         | 規則 70 条定義レコード ON のとき、line_id 列は 70 条用に払い出し<br>station_id は、b0～b15 の路線の計算駅<br>b0～15 は進入 or 脱出路線<br>→ 廃止(24 は空き)                                                                                                                                                         |
| 23          | BSRCOMPANY    | 会社線のとき 1                                                                                                                                                                                                                                                               |
| 22          | BSRVIRJCT     | 新幹線で新幹線の駅の無い在来線分岐駅(平行在来線の新幹線がない分岐駅。名古屋、沼津とか新富士は含まず、金山、草津とか山科は含む。また国府津、富士、沼津などは異なる線なので含まない。※ 分岐駅の列挙「乗車経路マーク」時に使用                                                                                                                                                        |
| 21          | BSRBORDER     | 本州 3 社と 3 島会社の運賃計算用の境界駅                                                                                                                                                                                                                                                |
| 20-17       | BSR69NOMASK   | 0～15 : 営業規則 69 条の項番に相当(経路 ID)t_rule69<br>※「在来線」で(BSR69TERM=1)の場合に適用                                                                                                                                                                                                    |
| 16-13       | BSRHZLIN      | 0～15: 新幹線の並行在来線 ID テーブル(t_hzline の rowid)<br>(新幹線のみ)<br>RouteItem 揃入時、0 に初期化され、以下の意味を持つ<br>16: 1 で経由表示時、路線非表示(中間アイテムのときのみ)<br>15: 1 で経由表示時、駅名非表示(中間アイテムのときのみ)<br>14: reserve<br>13: reserve                                                                            |
| BSR70=1 の場合 |               |                                                                                                                                                                                                                                                                        |

|      |           |                                  |
|------|-----------|----------------------------------|
| 15-0 | BSR70MASK | 大都市環状線(70条適用駅)エリアへの進入 or 脱出路線 id |
| 上記以外 |           |                                  |
| 12-0 | sflg の内容  |                                  |

```
#define FLGTEST(flg, MASK) (flg & MASK)
#define FLGVALU(flg, MASK, POS) ((flg & MASK) >> POS)
// 一般路線
#define BSRNMLMASK (~ (BSR70 | ...))
```

### city\_flg(sflg)の整理

| テーブル      |          | 列                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
|-----------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-------|------|----|------|------|---|-------|------|---|------|------|---|------|------|---|------|------|---|------|------|---|-------|------|---|------|------|----|------|------|----|------|------|
| t_station |          | cityflg                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| ビット       | ラベル      | 説明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| -13       |          | Lflg の内容                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 12        | BCJCTFLG | 0=分岐駅なし、1=分岐駅あり                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 11        | BCOSMSP  | 大阪電車特定区間                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 10        | BCTKMS   | 東京電車特定区間                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 7-9       | BCSUBURB | 近郊区間<br><table border="1" style="margin-left: 20px;"> <tr><td>1</td><td>東京</td></tr> <tr><td>2</td><td>新潟</td></tr> <tr><td>3</td><td>大阪</td></tr> <tr><td>4</td><td>福岡</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                            | 1 | 東京    | 2    | 新潟 | 3    | 大阪   | 4 | 福岡    |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 1         | 東京       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 2         | 新潟       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 3         | 大阪       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 4         | 福岡       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 6         | BCRULE70 | 旅客営業規則 70条適用駅                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 5         | BCYAMATE | 山手線内駅/大阪環状線内(大阪か山手かは BCOSMSP/BCTKMS 参照)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 4         | BCCITYCT | 都区市内中心駅の場合 1(BCCITYNO≠0 であること)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 0-3       | BCCITYNO | 都区市内番号(1~12)<br><table border="1" style="margin-left: 20px;"> <tr><td>1</td><td>東京都区内</td><td>0x01</td></tr> <tr><td>2</td><td>横浜市内</td><td>0x02</td></tr> <tr><td>3</td><td>名古屋市内</td><td>0x03</td></tr> <tr><td>4</td><td>京都市内</td><td>0x04</td></tr> <tr><td>5</td><td>大阪市内</td><td>0x05</td></tr> <tr><td>6</td><td>神戸市内</td><td>0x06</td></tr> <tr><td>7</td><td>広島市内</td><td>0x07</td></tr> <tr><td>8</td><td>北九州市内</td><td>0x08</td></tr> <tr><td>9</td><td>福岡市内</td><td>0x09</td></tr> <tr><td>10</td><td>仙台市内</td><td>0x0a</td></tr> <tr><td>11</td><td>札幌市内</td><td>0x0b</td></tr> </table> | 1 | 東京都区内 | 0x01 | 2  | 横浜市内 | 0x02 | 3 | 名古屋市内 | 0x03 | 4 | 京都市内 | 0x04 | 5 | 大阪市内 | 0x05 | 6 | 神戸市内 | 0x06 | 7 | 広島市内 | 0x07 | 8 | 北九州市内 | 0x08 | 9 | 福岡市内 | 0x09 | 10 | 仙台市内 | 0x0a | 11 | 札幌市内 | 0x0b |
| 1         | 東京都区内    | 0x01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 2         | 横浜市内     | 0x02                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 3         | 名古屋市内    | 0x03                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 4         | 京都市内     | 0x04                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 5         | 大阪市内     | 0x05                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 6         | 神戸市内     | 0x06                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 7         | 広島市内     | 0x07                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 8         | 北九州市内    | 0x08                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 9         | 福岡市内     | 0x09                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 10        | 仙台市内     | 0x0a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |
| 11        | 札幌市内     | 0x0b                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |       |      |    |      |      |   |       |      |   |      |      |   |      |      |   |      |      |   |      |      |   |       |      |   |      |      |    |      |      |    |      |      |

CITYFLG はいれられるものは spe\_route に含めることにより、t\_lines 参照時、t\_station を結合する手間を省く

(参考)

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |   |   |   |   |   |   |   |   |   |