

```
__code cs1 (*ids, *ods) {  
  sort (ids[0], ids[1]);  
  goto nextCode();  
}
```

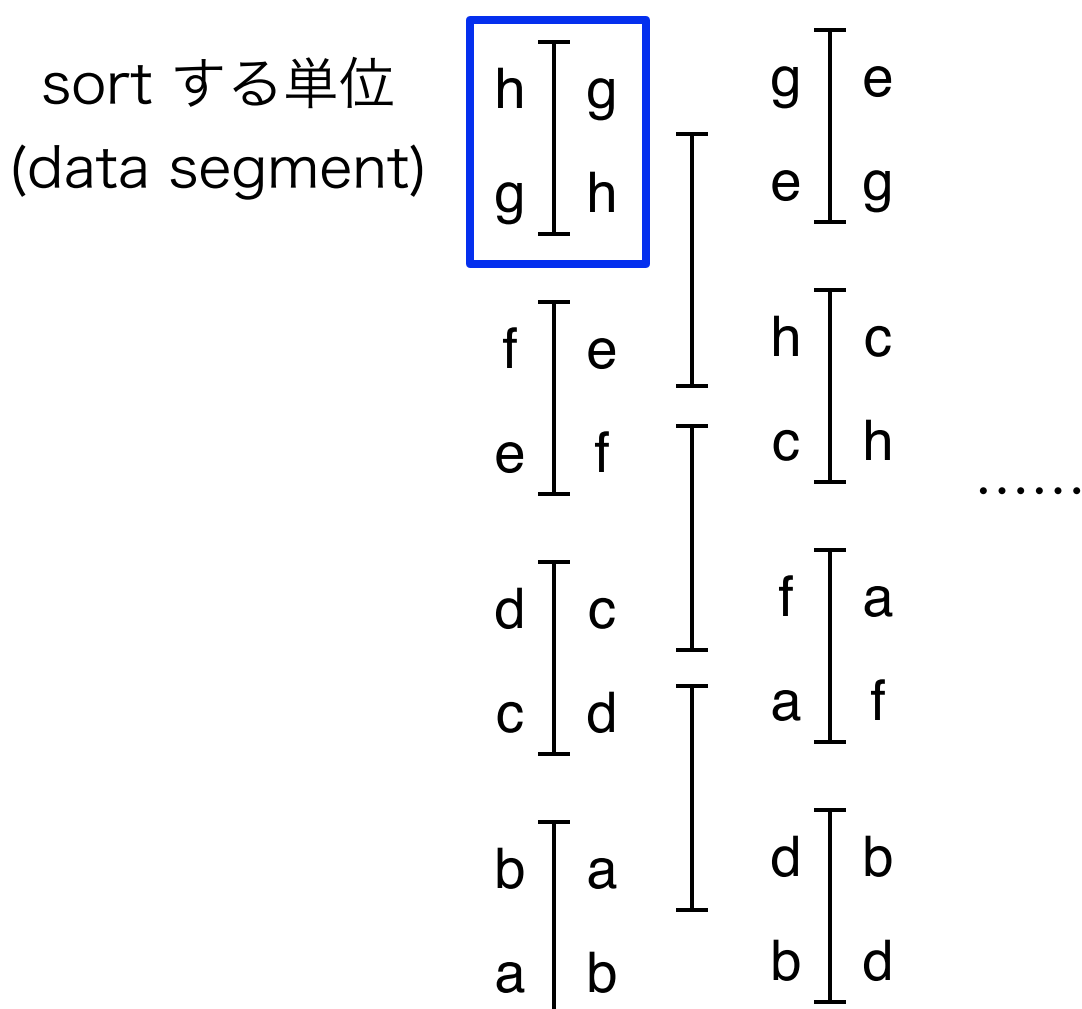
code segment と data segment の組が task となる。

実際には meta code segment に 継続する。

```
goto metaCode(*nextCode, *ods, *mds);
```

引数として継続先の code segment, output data segment, meta data segment を持つ。

bitonic sort での例



data segment の待ち合わせは meta code segment で制御する。code segment レベルでは wait を意識しない。

また、data segment の分割を許すかどうかという問題がある。許す場合は大きな data segment から一部分を取り出して code segment に与えることができる。この問題を bitonic sort の場合に当てはめると、sort する単位ごとに data segment を作るか sort 対象の配列1つが data segment になるかといったものになる。

input data segment と output data segment を入れ替える flip という機能がある。

ODSがIDSの書き換えの場合に用いることで無駄なデータのコピーを防ぐ。これは自動的に行われるのが良い。