

# 99 questions/Solutions/89

## From HaskellWiki

< 99 questions | Solutions

```
import Data.List
```

```
type Node = Int
```

```
type Edge = (Node,Node)
```

```
type Graph = ([Node],[Edge])
```

```
dfsbiptite :: Graph -> [(Node, Int)] -> [Node] -> [Node] -> Bool
```

```
dfsbiptite ([],_) _ _ = True
```

```
dfsbiptite (_,_) [] _ = True
```

```
dfsbiptite (v,e) ((nv, 0):stack) odd even
```

```
  | [x|x<-v,x==nv] == [] = dfsbiptite (v, e) stack odd even
```

```
  | [] == intersect adjacent even = dfsbiptite (newv, e) ([ (x,1) | x<-adjacent ] ++ stack) o
```

```
  | otherwise = False
```

```
where
```

```
  adjacent = [x | (x,y)<-e,y==nv] ++ [x | (y,x)<-e,y==nv]
```

```
  newv = [x|x<-v,x/=nv]
```

```
dfsbiptite (v,e) ((nv, 1):stack) odd even
```

```
  | [x|x<-v,x==nv] == [] = dfsbiptite (v, e) stack odd even
```

```
  | [] == intersect adjacent odd = dfsbiptite (newv, e) ([ (x,0) | x<-adjacent ] ++ stack) (n
```

```
  | otherwise = False
```

```
where
```

```
  adjacent = [x | (x,y)<-e,y==nv] ++ [x | (y,x)<-e,y==nv]
```

```
  newv = [x|x<-v,x/=nv]
```

```
biptite :: Graph -> Bool
```

```
biptite ([],_) = True
```

```
biptite (top:v,e) = dfsbiptite (top:v, e) [(top,0)] [] []
```

You can call it:

```
biptite ([1,2,3,4,5],[ (1,2), (2,3), (1,4), (3,4), (5,2), (5,4) ])
```

Retrieved from "[https://wiki.haskell.org/index.php?title=99\\_questions/Solutions/89&oldid=38926](https://wiki.haskell.org/index.php?title=99_questions/Solutions/89&oldid=38926)"

---

- This page was last modified on 5 March 2011, at 23:31.
- Recent content is available under a simple permissive license.