

ساختمان داده ها

درخت دودویی نخ کشی شده (Threaded Binary Tree)

مدرس: غیاثی شیرازی
دانشگاه فردوسی مشهد

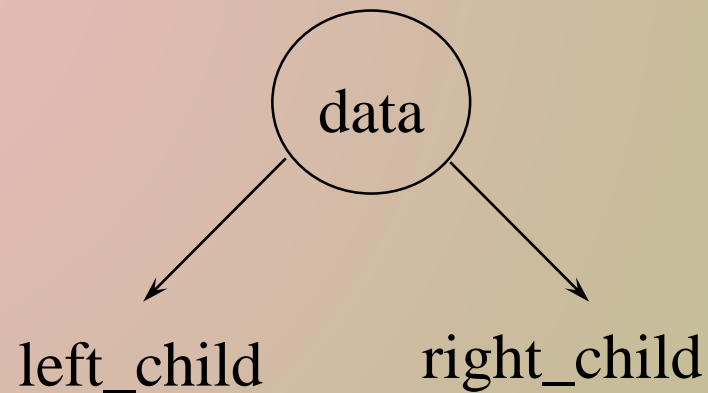
درخت دودویی نخ کشی شده

- در یک درخت دودویی با n گره تعداد $2n$ اشاره گر وجود دارد که $n+1$ تا از آنها null هستند.
- زیرا به هر گره غیر از ریشه دقیقاً یک اشاره گر اشاره می کند.
- ایده: در این اشاره گر ها مقادیر مفیدی را ذخیره نماییم.



ساختار گره ها در درخت دودویی معمولی

```
template <class T>
class TreeNode {
    TreeNode<T> *mLeftChild;
    TreeNode<T> *mRightChild;
    T mData;
};
```

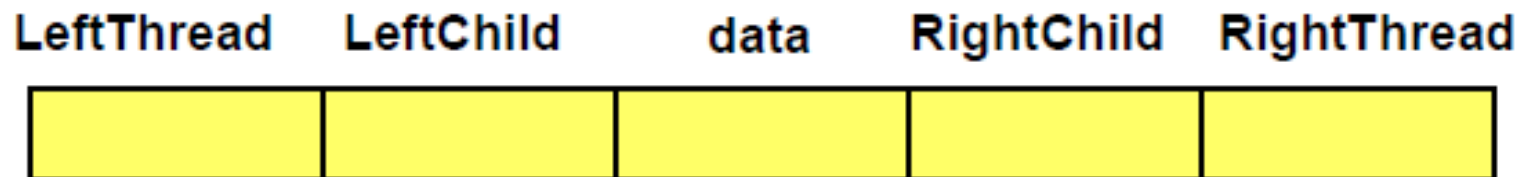


درخت دودویی نخ کشی شده

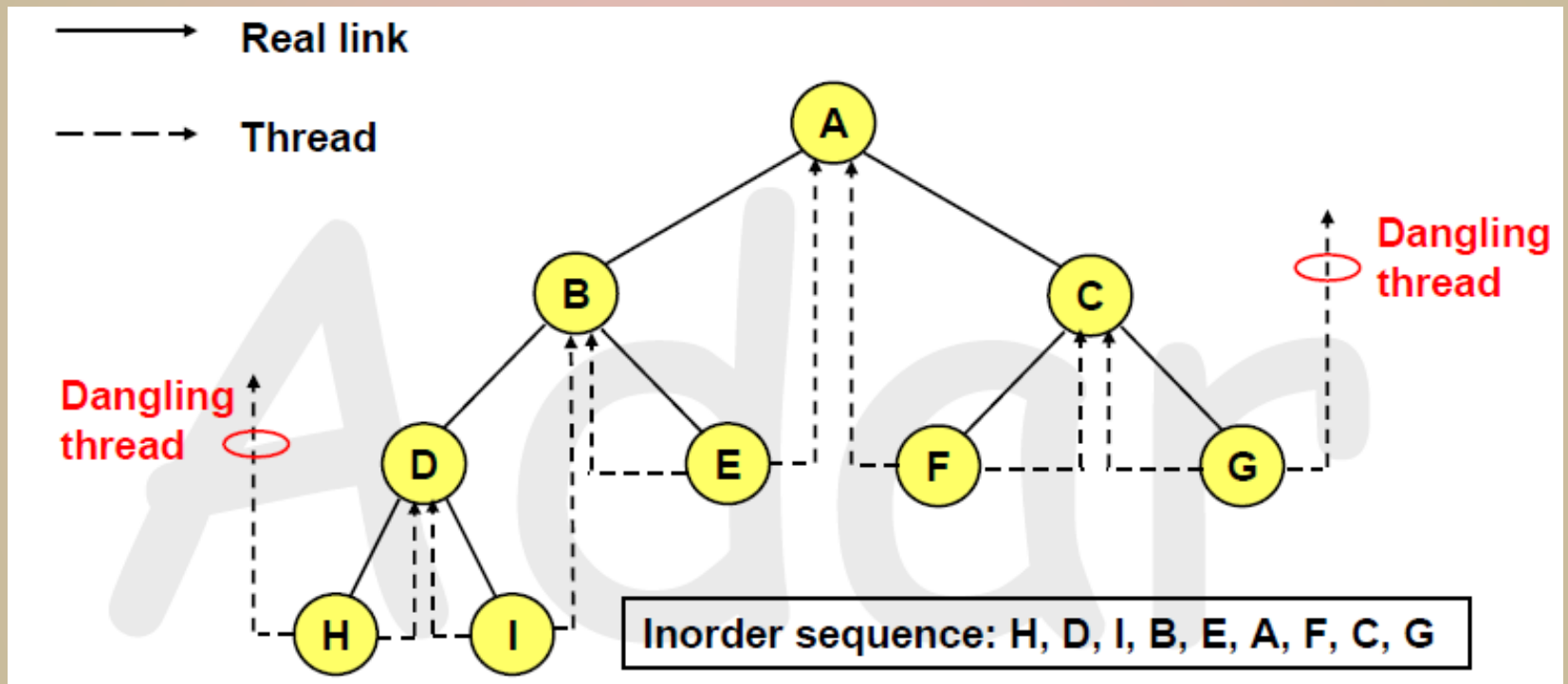
- اگر LeftChild گره p برابر null باشد، مقدار آن را برابر گره قبل از p در ترتیب inorder قرار می دهیم.
- اگر RightChild گره p برابر null باشد، مقدار آن را برابر گره بعد از p در ترتیب inorder قرار می دهیم.
- برای تفکیک اشاره گر معمولی با اشاره گرهای جدید متغیرهای منطقی LeftThread و RightThread را به ساختمان داده گره اضافه می کنیم.

ساختار گره ها در درخت دودویی نخ کشی شده

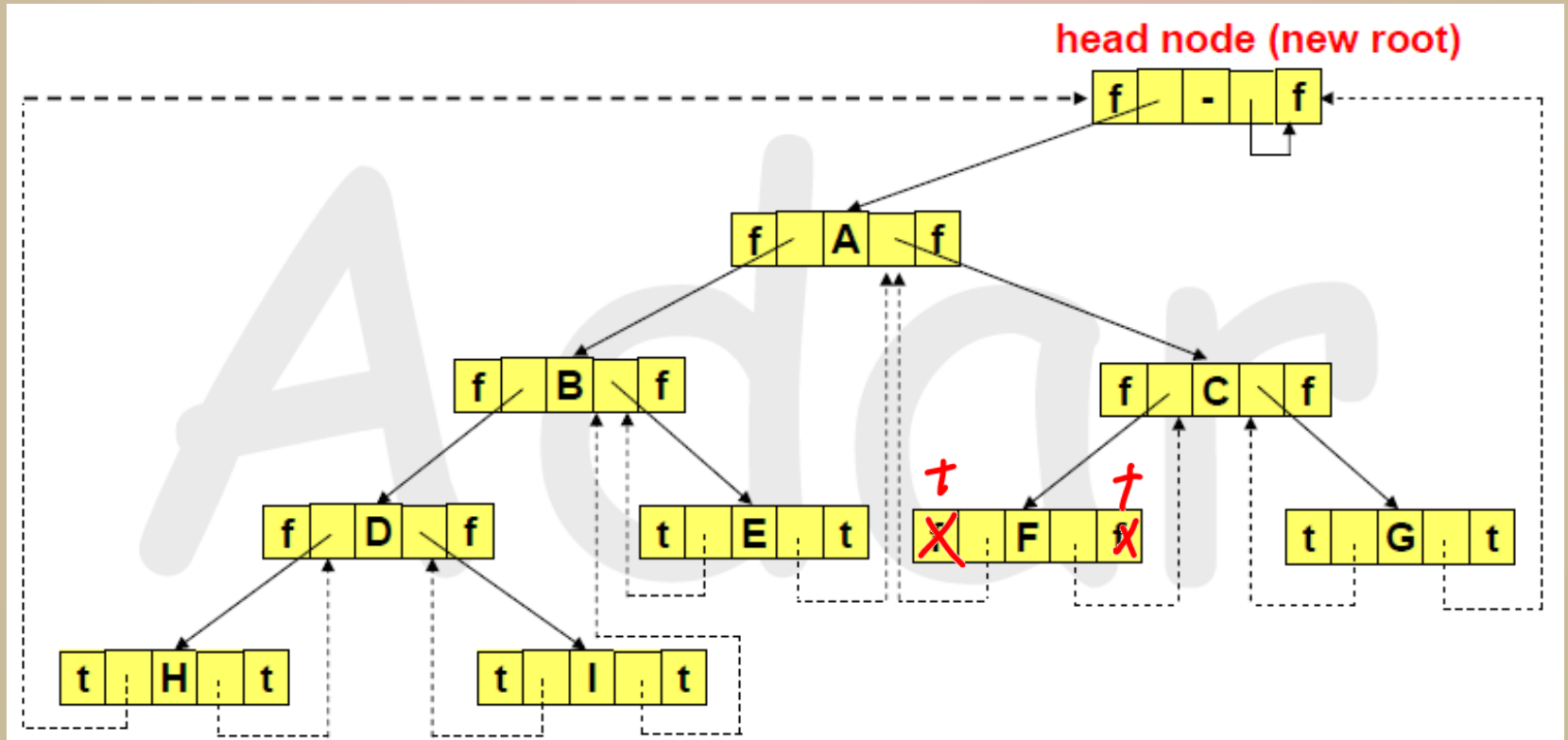
```
template <class T>
class TreeNode {
    bool            mLeftThread;
    bool            mRightThread;
    TreeNode<T> *mLeftChild;
    TreeNode<T> *mRightChild;
    T                mData;
};
```



مشکل نخ های سرگردان



حل مشکل با گره سرآیند اضافی



میزان مصرف حافظه

- با پیشرفت های بوجود آمده در تولید حافظه های با ظرفیت بالا، امروزه سرعت اهمیت بیشتری از حافظه دارد.
- هرچند یک متغیر منطقی را می توان در یک بیت ذخیره کرد، اما اندازه حافظه مصرفی یک کلاس باید مضرب صحیحی از بایت (۸ بیت) باشد.
- از آنجا که دسترسی به خانه هایی از حافظه که مضرب ۴ و ۸ هستند، سریع تر است، یک متغیر منطقی معمولا ۴ بایت مصرف می کند.

۸ بایت ، ۶۴ بیت

میزان مصرف حافظه

- بنابراین در طرح ارائه شده هیچ کاهشی در مصرف حافظه نداریم زیرا دو متغیر منطقی به اندازه دو اشاره گر حافظه مصرف می کنند (بلکه حافظه مصرفی افزایش یافته است).
- راه حل با هدف صرفه جویی در مصرف حافظه:
 - دو متغیر منطقی LeftThread و RightThread را به عنوان دو بیت از یک داده از نوع char ذخیره کنیم.
 - از فیلدهای بیتی در C++ استفاده کنیم.

ذخیره سازی متغیرهای منطقی به صورت بیتی

```
#pragma pack (1)
```

```
template <class T>
```

```
class ThreadedNode {
```

```
    ThreadedNode<T>*mLeftChild;
```

```
    ThreadedNode<T>*mRightChild;
```

```
    T                mData;
```

```
    unsigned int
```

```
    unsigned int
```

```
    mLeftThread : 1;
```

```
    mRightThread : 1;
```

```
};
```

یک بایت

بیت

جمع بندی بحث مصرف حافظه

- بحث صرفه جویی در مصرف حافظه در درخت های نخ کشی شده در بیشتر کاربردها اهمیتی ندارد.
- اما درخت های نخ کشی شده از این جهت که عملیات پیمایش درخت را سرعت می بخشند، اهمیت دارند.