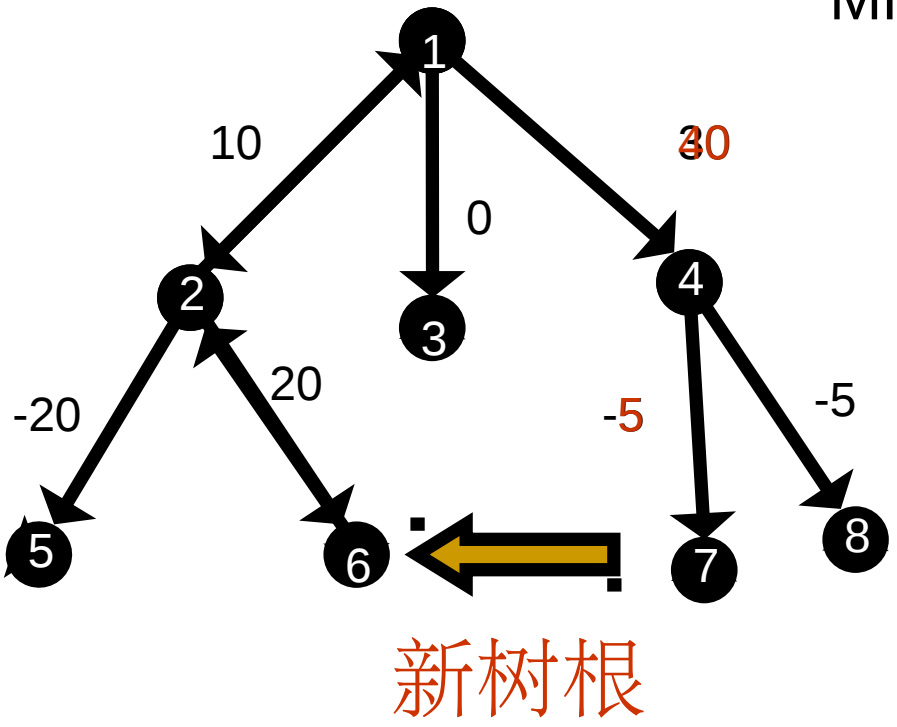
维护森林连通性——动态树

华东师大二附中
陈首元

动态树

- 维护一个森林
- 支持边的插入与删除
- 支持树的合并与分离
- 支持寻找路径上费用最小的边
- 所有操作的均摊复杂度为 O(logN)



Root(1,2,3,4,5,6,7,8)=1

MinCost(6)=10

Update(7,10)

Evert(6)

Cut(4)

Root(7,8)=4

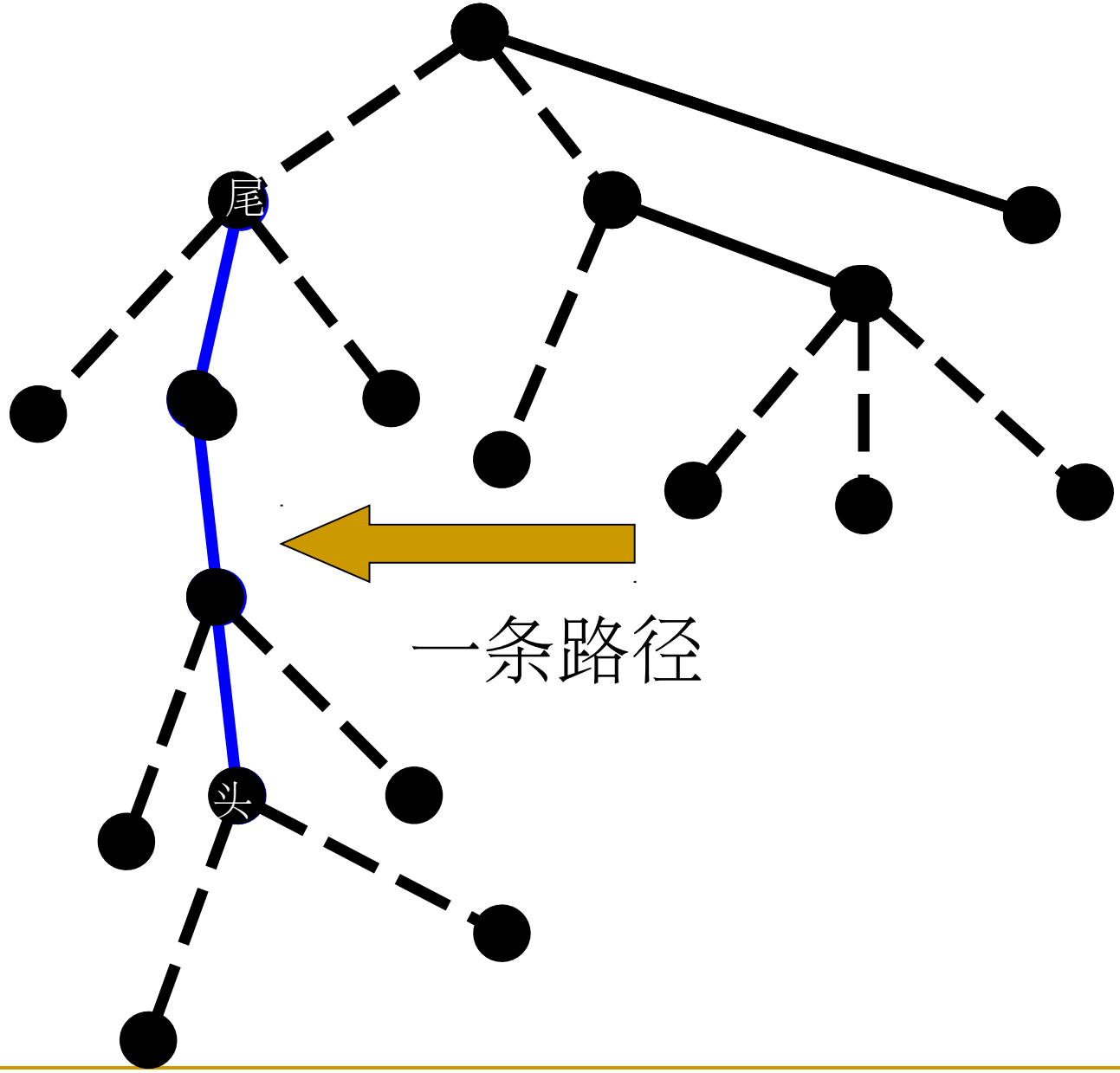
Link(3,4,10)

动态树的基本操作

- Root(v) 返回包含节点 v 的树的根
- MinCost(v) 返回 v 到根路径上费用最小的边
- Update(v,x) 使 v 到树根路径上的边的费用 +x
- Link(v,w,x) 将以 v 为根的树连接到节点 root(w) 上，（v,w）的费用为 x
- Cut(v) 删除 v 与其父节点连接的边
- Evert(v) 使 v 成为新的根，并将 v 到原树根上的边反向

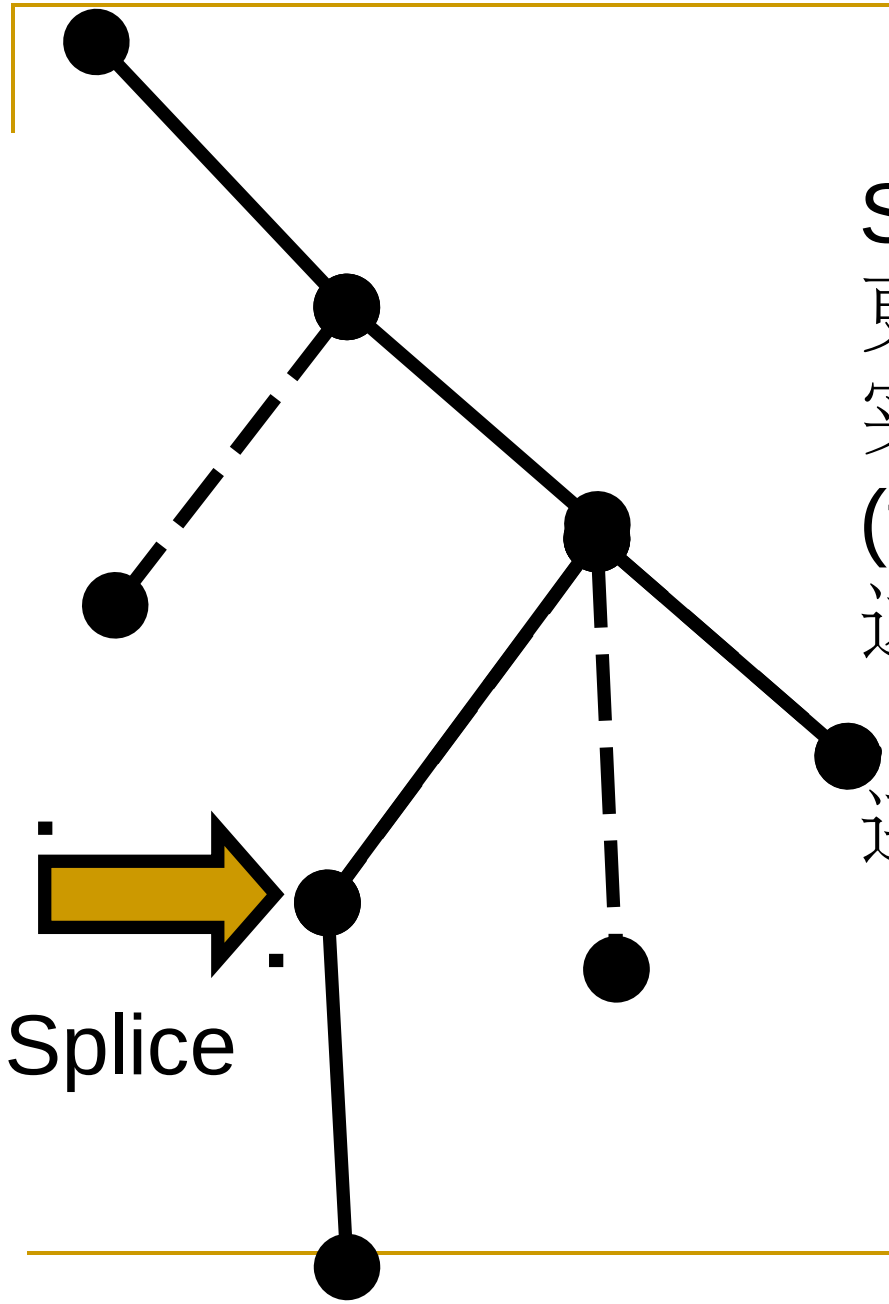
操作的实现

- 将树中的边分为实边、虚边两种，每个节点最多向其子节点连出一条实边
- 将树划分为一些完全由实边组成的路径，只对这些路径进行操作


尾
头
一条路径

路径的基本操作

- Path(v)： 返回包含 v 的路径 （每个路径有一个标志）
- Head(p),Tail(p)： 返回首节点、尾节点
- Pmincost(p)： 返回 p 中费用最小的边
- Pupdate(p,x:real)： 将 p 中每条边的费用 +x
- Reverse(p)： 将 p 中的每条边反向
- Concatenate(p,q,x) ： 添加边 (tail(p),head(q)）费用为 x ， 将路径 p,q 合并
- Split(v)： 将 v 从路径中删除并把路径分为两部分



Splice(p) ： 将路径 p 向更靠近根的方向增长

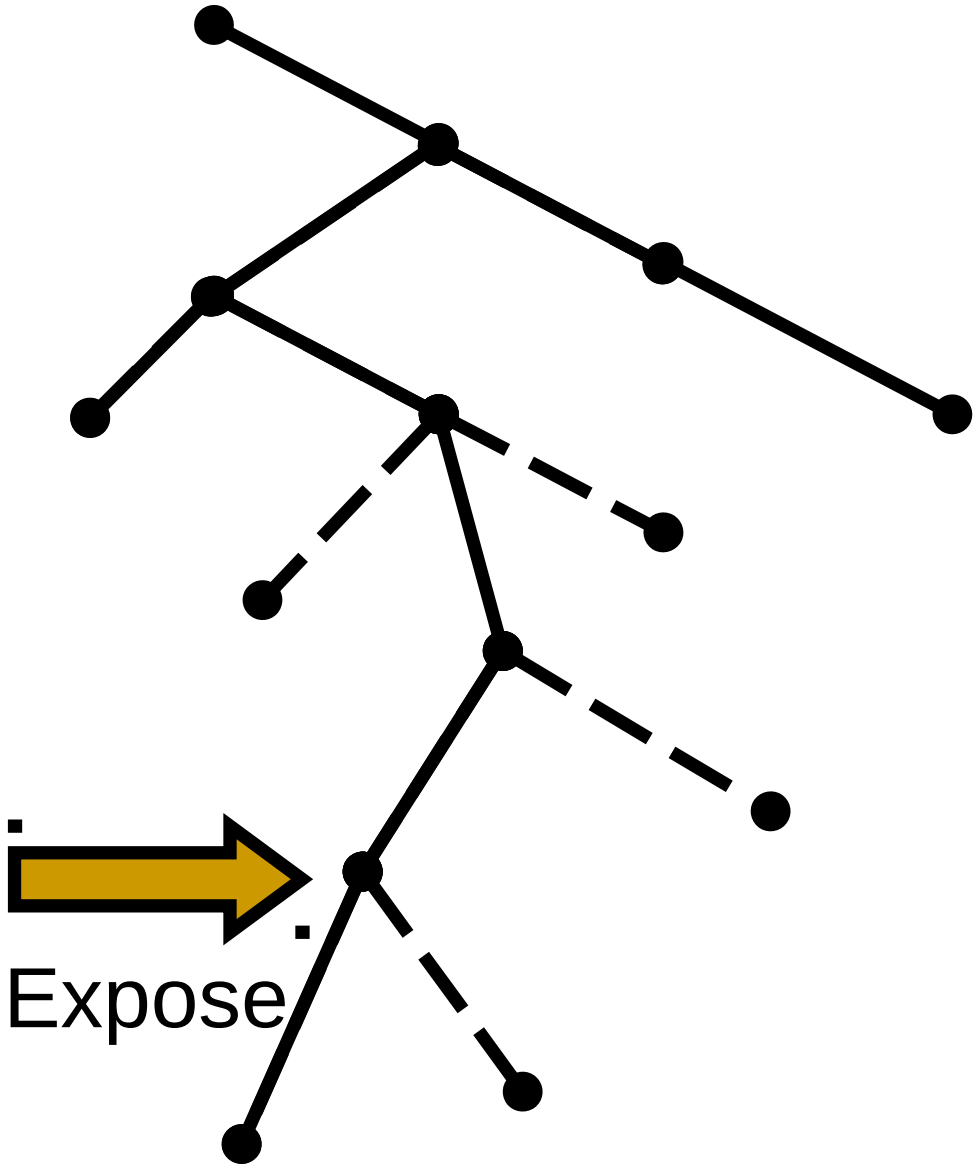
实现方法：把虚边(tail(P),Parent(p)) 变为实边，为了维护实边的性质

将原来从 Parent(P) 中连出的边设为虚边

Expose(v)：将从 v 到树根路径上的所有边设为实边

实现方法：不断调用 splice 直到根为止

有了 **Expose(v)**，就可以实现所有动态树操作了



Link 操作

- Procedure Link(v,w:vertex;x:real);
- Begin
- Concatenate(path(v),expose(w),x);
- End;

Cut 操作

- Function Cut(v:vertex);
- Var
- p,q:path;
- x,y:real;
- Begin
- Expose(v);
- [p,q,x,y]:=split(v);
- Dparent(v):=nil;
- Return y;
- End;

路径结构

- 用伸展树保存路径
- 所有路径操作都能用 Splay 实现
- 对于 Reverse 与 PUpdate 操作，可以采用先存放修改标志，在访问时修改的办法
- 在每次 Split 和 Concatenate 操作之后， Splay 最左子孙，使它成为根

复杂度分析（Expose 分析）

- 对于边 v->w （动态树中，不是伸展树中）如果 v 的子孙 >w 的子孙一半，称为 A 类边，否则成为 B 类边。
- 显然每个点最多连出一条 A 类边，树中每条路径上最多有 O(logN) 条 B 类边。
- 令 p 为 B 类边中的虚边数。

复杂度分析（Expose 分析）

- 一次 Expose 操作，将执行数次 Splice
- 对于每次 Splice：
  - 添加一条 B 类虚边进入路径： p+1
  - 这种情况最多发生 O(logN) 次
  - 添加一条 A 类虚边进入路径： p-1
  - 可能发生许多次，但代价是 p， p 是保持非负的。
- 由上面分析可以看出平均每次 expose 操作要执行 O(logN) 次路径操作

复杂度分析（Splay 分析）

- 势函数定义 r(x)： x 在动态树中的子节点数的对数
- 伸展树的性质：每一次 splay 操作的均摊复杂度不超过： 3(r(t)-r(x))+1
- 此性质在势函数定义改变后依然成立

复杂度分析

- Expose 的均摊复杂度 3(r(root)-r(v))+(v 到树根路径上的节点数)
- 由 Expose 分析得到路径上的节点数平均为 O(logN)
- 复杂度为 O(logN)

实现

- 不保存整棵树，而保存“虚拟树”。虚拟树中的顶点与动态树中的是一一对应的。虚拟树中的每个顶点最多连出两条实边，其余为虚边，分别成为左儿子和右儿子，其他顶点称为中间顶点。完全由实边组成的子树称为实树。

实现

- 实现 expose 需要两种操作：一种是 splay，将一棵实树（也是二叉树），从某个节点伸展使这个节点成为这个实树的根。第二种是 splice，将某个节点的一个中间节点变为左儿子，左儿子变为中间节点。
- 执行 expose(v) 操作需要分为三步。首先沿 v 往上到虚拟树的树根，对沿路的各实树进行 splay，执行完这个步后从 v 到树根的路径上只有虚边。再沿 v 往上到虚拟树的树根，对沿路各节点执行 splice，将从 v 到虚拟树树根路径上的边变为虚边。这时 v 和树根在一个实树中了。这时再执行 splay(v)，把 v 调整为树根。

谢谢