2. Text output of *gprof* with *callgraph* and code annotate

//PRN : 22610090

Flat profile:

Each sample counts as 0.01 seconds.

no time accumulated

% time	cumulative seconds	self seconds	calls	self Ts/call	total Ts/call	name
0.00	0.00	0.00	1010	0.00	0.00	__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::base() const
0.00	0.00	0.00	788	0.00	0.00	__gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type std::sqrt<int>(int)
0.00	0.00	0.00	600	0.00	0.00	__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator++()
0.00	0.00	0.00	600	0.00	0.00	__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator*() const
0.00	0.00	0.00	504	0.00	0.00	bool __gnu_cxx::operator!=<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&)
0.00	0.00	0.00	100	0.00	0.00	iseve(int)
0.00	0.00	0.00	100	0.00	0.00	isprm(int)
0.00	0.00	0.00	100	0.00	0.00	std::vector<long long, std::allocator<long long> >::operator[](unsigned long)
0.00	0.00	0.00	11	0.00	0.00	__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::__normal_iterator(long long* const&)

0.00 0.00 0.00 6 0.00 0.00 std::__new_allocator<long long>::~__new_allocator()

0.00 0.00 0.00 6 0.00 0.00 std::vector<long long, std::allocator<long long> >::begin()

0.00 0.00 0.00 5 0.00 0.00 std::vector<long long, std::allocator<long long> >::end()

0.00 0.00 0.00 4 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_M_get_Tp_allocator()

0.00 0.00 0.00 2 0.00 0.00 std::vector<long long, std::allocator<long long> >::size() const

0.00 0.00 0.00 2 0.00 0.00 void std::_Destroy_aux<true>::__destroy<long long*> (long long*, long long*)

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_M_allocate(unsigned long)

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::_Vector_impl(std::allocator<long long> const&)

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::~_Vector_impl()

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_M_deallocate(long long*, unsigned long)

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_M_create_storage(unsigned long)

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl_data::_Vector_impl_data()

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::_Vector_base(unsigned long, std::allocator<long long> const&)

0.00 0.00 0.00 2 0.00 0.00 std::_Vector_base<long long, std::allocator<long long> >::~_Vector_base()

0.00 0.00 0.00 2 0.00 0.00 std::__new_allocator<long long>::deallocate(long long*, unsigned long)

0.00 0.00 0.00 2 0.00 0.00 std::__new_allocator<long long>::allocate(unsigned long, void const*)

0.00 0.00 0.00 2 0.00 0.00 long long* std::__uninitialized_default_n_1<true>::__uninit_default_n<long long*, unsigned long>(long long*, unsigned long)

0.00 0.00 0.00 2 0.00 0.00 std::vector<long long, std::allocator<long long> >::_S_max_size(std::allocator<long long> const&)

0.00 0.00 0.00 2 0.00 0.00 std::vector<long long, std::allocator<long long> >::_S_check_init_len(unsigned long, std::allocator<long long> const&)

0.00 0.00 0.00 2 0.00 0.00 std::vector<long long, std::allocator<long long> >::_M_default_initialize(unsigned long)

0.00 0.00 0.00 2 0.00 0.00 std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&)

0.00 0.00 0.00 2 0.00 0.00 std::vector<long long, std::allocator<long long> >::~vector()

0.00 0.00 0.00 2 0.00 0.00 void std::_Construct<long long>(long long*)

0.00 0.00 0.00 2 0.00 0.00 long long* std::__fill_n_a<long long*, unsigned long, long long>(long long*, unsigned long, long long const&, std::random_access_iterator_tag)

0.00 0.00 0.00 2 0.00 0.00 long long* std::__addressof<long long>(long long&)

0.00 0.00 0.00 2 0.00 0.00 std::__size_to_integer(unsigned long)

0.00 0.00 0.00 2 0.00 0.00 long long* std::__uninitialized_default_n<long long*, unsigned long>(long long*, unsigned long)

0.00 0.00 0.00 2 0.00 0.00 long long* std::__uninitialized_default_n_a<long long*, unsigned long, long long>(long long*, unsigned long, std::allocator<long long>&)

0.00 0.00 0.00 2 0.00 0.00 unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&)

0.00 0.00 0.00 2 0.00 0.00 long long* std::fill_n<long long*, unsigned long, long long>(long long*, unsigned long, long long const&)

0.00 0.00 0.00 2 0.00 0.00 void std::_Destroy<long long*>(long long*, long long *)

0.00 0.00 0.00 2 0.00 0.00 void std::__fill_a<long long*, long long>(long long*, long long*, long long const&)

0.00 0.00 0.00 2 0.00 0.00 __gnu_cxx::__enable_if<std::__is_scalar<long long>::__value, void>::__type std::__fill_a1<long long*, long long>(long long*, long long*, long

long const&)

0.00 0.00 0.00 2 0.00 0.00 operator new(unsigned long, void*)

0.00 0.00 0.00 1 0.00 0.00 average(std::vector<long long, std::allocator<long long> >&)

0.00 0.00 0.00 1 0.00 0.00 prefsum(std::vector<long long, std::allocator<long long> >&)

0.00 0.00 0.00 1 0.00 0.00 randomnum(long long, long long, long long)

0.00 0.00 0.00 1 0.00 0.00 bool __gnu_cxx::operator==<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&)

0.00 0.00 0.00 1 0.00 0.00 double std::accumulate<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double>(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double)

0.00 0.00 0.00 1 0.00 0.00 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >)

% the percentage of the total running time of the

time program used by this function.

cumulative a running sum of the number of seconds accounted

seconds for by this function and those listed above it.

self the number of seconds accounted for by this

seconds function alone. This is the major sort for this
listing.

calls the number of times this function was invoked, if
this function is profiled, else blank.

self the average number of milliseconds spent in this
ms/call function per call, if this function is profiled,
else blank.

total the average number of milliseconds spent in this
ms/call function and its descendents per call, if this
function is profiled, else blank.

name the name of the function. This is the minor sort
for this listing. The index shows the location of
the function in the gprof listing. If the index is
in parenthesis it shows where it would appear in
the gprof listing if it were to be printed.

Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

index % time self children called name

0.00 0.00 2/1010 bool __gnu_cxx::operator==<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [54]

0.00 0.00 1008/1010 bool __gnu_cxx::operator!=<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [12]

[8] 0.0 0.00 0.00 1010 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::base() const [8]

0.00 0.00 788/788 isprm(int) [14]

[9] 0.0 0.00 0.00 788 __gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type std::sqrt<int>(int) [9]

0.00 0.00 100/600 double std::accumulate<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double>(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double) [55]

0.00 0.00 200/600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >) [56]

0.00 0.00 300/600 main [6]

[10] 0.0 0.00 0.00 600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator++() [10]

0.00 0.00 100/600 double std::accumulate<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double>(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double) [55]

0.00 0.00 200/600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >) [56]

0.00 0.00 300/600 main [6]

[11] 0.0 0.00 0.00 600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator*() const [11]

0.00 0.00 100/504 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >) [56]

0.00 0.00 101/504 double std::accumulate<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double> (__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double) [55]

0.00 0.00 303/504 main [6]

[12] 0.0 0.00 0.00 504 bool __gnu_cxx::operator!=<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long

long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [12]

0.00 0.00 1008/1010 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::base() const [8]

0.00 0.00 100/100 main [6]

[13] 0.0 0.00 0.00 100 iseve(int) [13]

0.00 0.00 100/100 main [6]

[14] 0.0 0.00 0.00 100 isprm(int) [14]

0.00 0.00 788/788 __gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type std::sqrt<int>(int) [9]

0.00 0.00 100/100 randomnum(long long, long long, long long) [53]

[15] 0.0 0.00 0.00 100 std::vector<long long, std::allocator<long long> >::operator[](unsigned long) [15]

0.00 0.00 5/11 std::vector<long long, std::allocator<long long> >::end() [19]

0.00 0.00 6/11 std::vector<long long, std::allocator<long long> >::begin() [18]

[16] 0.0 0.00 0.00 11 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::__normal_iterator(long long* const&) [16]

0.00 0.00 1/6 prefsum(std::vector<long long, std::allocator<long long> >&) [52]

0.00 0.00 1/6 randomnum(long long, long long, long long) [53]

0.00 0.00 2/6 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::~_Vector_impl() [25]

0.00 0.00 2/6 std::vector<long long, std::allocator<long long> >::_S_check_init_len(unsigned long, std::allocator<long long> const&) [35]

[17] 0.0 0.00 0.00 6 std::__new_allocator<long long>::~__new_allocator() [17]

0.00 0.00 1/6 average(std::vector<long long, std::allocator<long long> >&) [51]

[52] 0.00 0.00 2/6 prefsum(std::vector<long long, std::allocator<long long> >&)

[18] 0.00 0.00 3/6 main [6]

0.0 0.00 0.00 6 std::vector<long long, std::allocator<long long> >::begin() [18]

0.00 0.00 6/11 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::__normal_iterator(long long* const&) [16]

0.00 0.00 1/5 prefsum(std::vector<long long, std::allocator<long long> >&) [52]

[51] 0.00 0.00 1/5 average(std::vector<long long, std::allocator<long long> >&)

[19] 0.00 0.00 3/5 main [6]

0.0 0.00 0.00 5 std::vector<long long, std::allocator<long long> >::end() [19]

0.00 0.00 5/11 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::__normal_iterator(long long* const&) [16]

0.00 0.00 2/4 std::vector<long long, std::allocator<long long> >::~vector() [38]

0.00 0.00 2/4 std::vector<long long, std::allocator<long long> >::_M_default_initialize(unsigned long) [36]

[20] 0.0 0.00 0.00 4 std::_Vector_base<long long, std::allocator<long long> >::_M_get_Tp_allocator() [20]

0.00 0.00 1/2 prefsum(std::vector<long long, std::allocator<long long> >&) [52]

0.00 0.00 1/2 average(std::vector<long long, std::allocator<long long> >&) [51]

[21] 0.0 0.00 0.00 2 std::vector<long long, std::allocator<long long> >::size() const [21]

0.00 0.00 2/2 void std::_Destroy<long long*>(long long*, long long*) [47]

[22] 0.0 0.00 0.00 2 void std::_Destroy_aux<true>::__destroy<long long*>(long long*, long long*) [22]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_M_create_storage(unsigned long) [27]

[23] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::_M_allocate(unsigned long) [23]

0.00 0.00 2/2 std::__new_allocator<long long>::allocate(unsigned long, void const*) [32]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_base(unsigned long, std::allocator<long long> const&) [29]

[24] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::_Vector_impl(std::allocator<long long> const&) [24]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl_data::_Vector_impl_data() [28]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::~_Vector_base() [30]

[25] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::~_Vector_impl() [25]

0.00 0.00 2/6 std::__new_allocator<long long>::~__new_allocator() [17]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::~_Vector_base() [30]

[26] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::_M_deallocate(long long*, unsigned long) [26]

0.00 0.00 2/2 std::__new_allocator<long long>::deallocate(long long*, unsigned long) [31]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_base(unsigned long, std::allocator<long long> const&) [29]

[27] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::_M_create_storage(unsigned long) [27]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_M_allocate(unsigned long) [23]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::_Vector_impl(std::allocator<long long> const&) [24]

[28] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl_data::_Vector_impl_data() [28]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&) [37]

[29] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_base(unsigned long, std::allocator<long long> const&) [29]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::_Vector_impl(std::allocator<long long> const&) [24]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_M_create_storage(unsigned long) [27]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::~vector() [38]

[30] 0.0 0.00 0.00 2 std::_Vector_base<long long, std::allocator<long long> >::~_Vector_base() [30]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_M_deallocate(long long*, unsigned long) [26]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::~_Vector_impl() [25]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_M_deallocate(long long*, unsigned long) [26]

[31] 0.0 0.00 0.00 2 std::__new_allocator<long long>::deallocate(long long*, unsigned long) [31]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_M_allocate(unsigned long) [23]

[32] 0.0 0.00 0.00 2 std::__new_allocator<long long>::allocate(unsigned long, void const*) [32]

0.00 0.00 2/2 long long* std::__uninitialized_default_n<long long*, unsigned long>(long long*, unsigned long) [43]

[33] 0.0 0.00 0.00 2 long long* std::__uninitialized_default_n_1<true>::__uninit_default_n<long long*, unsigned long>(long long*, unsigned long) [33]

0.00 0.00 2/2 long long* std::__addressof<long long>(long long&) [41]

0.00 0.00 2/2 void std::_Construct<long long>(long long*) [39]

0.00 0.00 2/2 long long* std::fill_n<long long*, unsigned long, long long>(long long*, unsigned long, long long const&) [46]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::_S_check_init_len(unsigned long, std::allocator<long long> const&) [35]

[34] 0.0 0.00 0.00 2 std::vector<long long, std::allocator<long long> >::_S_max_size(std::allocator<long long> const&) [34]

0.00 0.00 2/2 unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&) [45]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&) [37]

[35] 0.0 0.00 0.00 2 std::vector<long long, std::allocator<long long> >::_S_check_init_len(unsigned long, std::allocator<long long> const&) [35]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::_S_max_size(std::allocator<long long> const&) [34]

0.00 0.00 2/6 std::__new_allocator<long long>::~__new_allocator() [17]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&) [37]

[36] 0.0 0.00 0.00 2 std::vector<long long, std::allocator<long long> >::_M_default_initialize(unsigned long) [36]

0.00 0.00 2/4 std::_Vector_base<long long, std::allocator<long long> >::_M_get_Tp_allocator() [20]

0.00 0.00 2/2 long long* std::__uninitialized_default_n_a<long long*, unsigned long, long long>(long long*, unsigned long, std::allocator<long long>&) [44]

0.00 0.00 1/2 prefsum(std::vector<long long, std::allocator<long long> >&) [52]

0.00 0.00 1/2 randomnum(long long, long long, long long) [53]

[37] 0.0 0.00 0.00 2 std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&) [37]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::_S_check_init_len(unsigned long, std::allocator<long long> const&) [35]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::_Vector_base(unsigned long, std::allocator<long long> const&) [29]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::_M_default_initialize(unsigned long) [36]

0.00 0.00 2/2 main [6]

[38] 0.0 0.00 0.00 2 std::vector<long long, std::allocator<long long> >::~vector() [38]

0.00 0.00 2/4 std::_Vector_base<long long, std::allocator<long long> >::_M_get_Tp_allocator() [20]

0.00 0.00 2/2 std::_Vector_base<long long, std::allocator<long long> >::~_Vector_base() [30]

0.00 0.00 2/2 void std::_Destroy<long long*>(long long*, long long*) [47]

0.00 0.00 2/2 long long* std::__uninitialized_default_n_1<true>::__uninit_default_n<long long*, unsigned long>(long long*, unsigned long) [33]

[39] 0.0 0.00 0.00 2 void std::_Construct<long long>(long long*) [39]

0.00 0.00 2/2 operator new(unsigned long, void*) [50]

0.00 0.00 2/2 long long* std::fill_n<long long*, unsigned long, long long>(long long*, unsigned long, long long const&) [46]

[40] 0.0 0.00 0.00 2 long long* std::__fill_n_a<long long*, unsigned long, long long>(long long*, unsigned long, long long const&, std::random_access_iterator_tag) [40]

0.00 0.00 2/2 void std::__fill_a<long long*, long long>(long long*, long long*, long long const&) [48]

0.00 0.00 2/2 long long* std::__uninitialized_default_n_1<true>::__uninit_default_n<long long*, unsigned long>(long long*, unsigned long) [33]

[41] 0.0 0.00 0.00 2 long long* std::__addressof<long long>(long long&) [41]

0.00 0.00 2/2 long long* std::fill_n<long long*, unsigned long, long long>(long long*, unsigned long, long long const&) [46]

[42] 0.0 0.00 0.00 2 std::__size_to_integer(unsigned long) [42]

0.00 0.00 2/2 long long* std::__uninitialized_default_n_a<long long*, unsigned long, long long>(long long*, unsigned long, std::allocator<long long>&) [44]

[43] 0.0 0.00 0.00 2 long long* std::__uninitialized_default_n<long long*, unsigned long>(long long*, unsigned long) [43]

0.00 0.00 2/2 long long* std::__uninitialized_default_n_1<true>::__uninit_default_n<long long*, unsigned long>(long long*, unsigned long) [33]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::_M_default_initialize(unsigned long) [36]

[44] 0.0 0.00 0.00 2 long long* std::__uninitialized_default_n_a<long long*, unsigned long, long long>(long long*, unsigned long, std::allocator<long long>&) [44]

0.00 0.00 2/2 long long* std::__uninitialized_default_n<long long*, unsigned long>(long long*, unsigned long) [43]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::_S_max_size(std::allocator<long long> const&) [34]

[45] 0.0 0.00 0.00 2 unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&) [45]

0.00 0.00 2/2 long long* std::__uninitialized_default_n_1<true>::__uninit_default_n<long long*, unsigned long>(long long*, unsigned long) [33]

[46] 0.0 0.00 0.00 2 long long* std::fill_n<long long*, unsigned long, long long>(long long*, unsigned long, long long const&) [46]

0.00 0.00 2/2 std::__size_to_integer(unsigned long) [42]

0.00 0.00 2/2 long long* std::__fill_n_a<long long*, unsigned long, long long>(long long*, unsigned long, long long const&, std::random_access_iterator_tag) [40]

0.00 0.00 2/2 std::vector<long long, std::allocator<long long> >::~vector() [38]

[47] 0.0 0.00 0.00 2 void std::_Destroy<long long*>(long long*, long long*) [47]

0.00 0.00 2/2 void std::_Destroy_aux<true>::__destroy<long long*>(long long*, long long*) [22]

0.00 0.00 2/2 long long* std::__fill_n_a<long long*, unsigned long, long long>(long long*, unsigned long, long long const&, std::random_access_iterator_tag) [40]

[48] 0.0 0.00 0.00 2 void std::__fill_a<long long*, long long>(long long*, long long*, long long const&) [48]

0.00 0.00 2/2 __gnu_cxx::__enable_if<std::__is_scalar<long long>::__value, void>::__type std::__fill_a1<long long*, long long>(long long*, long long*, long long const&) [49]

0.00 0.00 2/2 void std::__fill_a<long long*, long long>(long long*, long long*, long long const&) [48]

[49] 0.0 0.00 0.00 2 __gnu_cxx::__enable_if<std::__is_scalar<long long>::__value, void>::__type std::__fill_a1<long long*, long long>(long long*, long long*, long long const&) [49]

0.00 0.00 2/2 void std::_Construct<long long>(long long*) [39]

[50] 0.0 0.00 0.00 2 operator new(unsigned long, void*) [50]

0.00 0.00 1/1 main [6]

[51] 0.0 0.00 0.00 1 average(std::vector<long long, std::allocator<long long> >&) [51]

0.00 **0.00** 1/5 std::vector<long long, std::allocator<long long> >::end() [19]

0.00 0.00 1/6 std::vector<long long, std::allocator<long long> >::begin() [18]

1/1 double std::accumulate<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double>(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >,

__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double) [55]

0.00 0.00 1/2 std::vector<long long, std::allocator<long long> >::size() const [21]

0.00 0.00 1/1 main [6]

[52] 0.0 0.00 0.00 1 prefsum(std::vector<long long, std::allocator<long long> >&) [52]

0.00 0.00 2/6 std::vector<long long, std::allocator<long long> >::begin() [18]

[21] 0.00 0.00 1/2 std::vector<long long, std::allocator<long long> >::size() const

0.00 0.00 1/2 std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&) [37]

0.00 0.00 1/6 std::__new_allocator<long long>::~__new_allocator() [17]

0.00 0.00 1/5 std::vector<long long, std::allocator<long long> >::end() [19]

0.00 0.00 1/1 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >) [56]

0.00 0.00 1/1 main [6]

[53] 0.0 0.00 0.00 1 randomnum(long long, long long, long long) [53]

0.00 0.00 100/100 std::vector<long long, std::allocator<long long> >::operator[](unsigned long) [15]

0.00 0.00 1/2 std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&) [37]

0.00 0.00 1/6 std::__new_allocator<long long>::~__new_allocator() [17]

0.00 0.00 1/1 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >) [56]

[54] 0.0 0.00 0.00 1 bool __gnu_cxx::operator==<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [54]

0.00 0.00 2/1010 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::base() const [8]

0.00 0.00 1/1 average(std::vector<long long, std::allocator<long long> >&) [51]
[55] 0.0 0.00 0.00 1 double std::accumulate<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double>(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double) [55]

0.00 0.00 101/504 bool __gnu_cxx::operator!=<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [12]

0.00 0.00 100/600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator*() const [11]

0.00 0.00 100/600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator++() [10]

0.00 0.00 1/1 prefsum(std::vector<long long, std::allocator<long long> >&) [52]
[56] 0.0 0.00 0.00 1 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*,

std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >) [56]

0.00 0.00 200/600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator*() const [11]

0.00 0.00 200/600 __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator++() [10]

0.00 0.00 100/504 bool __gnu_cxx::operator!=<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [12]

0.00 0.00 1/1 bool __gnu_cxx::operator==<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [54]

This table describes the call tree of the program, and was sorted by

the total amount of time spent in each function and its children.

Each entry in this table consists of several lines. The line with the

index number at the left hand margin lists the current function.

The lines above it list the functions that called this function,

and the lines below it list the functions this one called.

This line lists:

index A unique number given to each element of the table.

Index numbers are sorted numerically.

The index number is printed next to every function name so

it is easier to look up where the function is in the table.

% timeThis is the percentage of the `total' time that was spent
in this function and its children. Note that due to
di erent viewpoints, functions excluded by options, etc,
these numbers will NOT add up to 100%.

self This is the total amount of time spent in this function.

childrenThis is the total amount of time propagated into this
function by its children.

calledThis is the number of times the function was called.
If the function called itself recursively, the number
only includes non-recursive calls, and is followed by
a `+' and the number of recursive calls.

name The name of the current function. The index number is
printed after it. If the function is a member of a
cycle, the cycle number is printed between the
function's name and the index number.

For the function's parents, the fields have the following meanings:

self This is the amount of time that was propagated directly

from the function into this parent.

childrenThis is the amount of time that was propagated from the function's children into this parent.

calledThis is the number of times this parent called the function `/' the total number of times the function was called. Recursive calls to the function are not included in the number after the `/'.

name This is the name of the parent. The parent's index number is printed after it. If the parent is a member of a cycle, the cycle number is printed between the name and the index number.

If the parents of the function cannot be determined, the word `<spontaneous>' is printed in the `name' field, and all the other fields are blank.

For the function's children, the fields have the following meanings:

self This is the amount of time that was propagated directly from the child into the function.

childrenThis is the amount of time that was propagated from the

child's children to the function.

called This is the number of times the function called this child `/' the total number of times the child was called. Recursive calls by the child are not listed in the number after the `/'.

name This is the name of the child. The child's index number is printed after it. If the child is a member of a cycle, the cycle number is printed between the name and the index number.

If there are any cycles (circles) in the call graph, there is an entry for the cycle-as-a-whole. This entry shows who called the cycle (as parents) and the members of the cycle (as children.) The `+' recursive calls entry shows the number of function calls that were internal to the cycle, and the calls entry for each member shows, for that member, how many times it was called from other members of the cycle.

Index by function name

[13] iseve(int) [27] std::_Vector_base<long long, std::allocator<long long> >::_M_create_storage(unsigned long) [39] void std::_Construct<long long>(long long*)

[14] isprm(int) [28] std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl_data::_Vector_impl_data() [40] long long* std::__fill_n_a<long long*, unsigned long, long long>(long long*, unsigned long, long long const&, std::random_access_iterator_tag)

[51] average(std::vector<long long, std::allocator<long long> >&) [20] std::_Vector_base<long long, std::allocator<long long> >::_M_get_Tp_allocator() [55] double std::accumulate<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double>(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, double)
[52] prefsum(std::vector<long long, std::allocator<long long> >&) [29] std::_Vector_base<long long, std::allocator<long long> >::_Vector_base(unsigned long, std::allocator<long long> const&) [41] long long* std::__addressof<long long>(long long&)

[53] randomnum(long long, long long, long long) [30] std::_Vector_base<long long, std::allocator<long long> >::~_Vector_base() [56] __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > std::partial_sum<__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >)

[16] __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::__normal_iterator(long long* const&) [31] std::__new_allocator<long long>::deallocate(long long*, unsigned long) [42] std::__size_to_integer(unsigned long)

[10] __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator++() [32] std::__new_allocator<long long>::allocate(unsigned long, void const*) [43] long long* std::__uninitialized_default_n<long long*, unsigned long>(long long*, unsigned long)
[54] bool __gnu_cxx::operator==<long long*, std::vector<long long, std::allocator<long long> > >(__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [17] std::__new_allocator<long long>::~__new_allocator() [44] long long* std::__uninitialized_default_n_a<long long*, unsigned long, long long>(long long*, unsigned long, std::allocator<long long>&)

[12] bool __gnu_cxx::operator!=<long long*, std::vector<long long, std::allocator<long long> > > (__gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&, __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > > const&) [33] long long* std::__uninitialized_default_n_1<true>::__uninit_default_n<long long*, unsigned long>(long long*, unsigned long) [45] unsigned long const& std::min<unsigned long>(unsigned long const&, unsigned long const&)

[8] __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::base() const [34] std::vector<long long, std::allocator<long long> >::_S_max_size(std::allocator<long long> const&) [9] __gnu_cxx::__enable_if<std::__is_integer<int>::__value, double>::__type std::sqrt<int>(int)
[11] __gnu_cxx::__normal_iterator<long long*, std::vector<long long, std::allocator<long long> > >::operator*() const [35] std::vector<long long, std::allocator<long long> >::_S_check_init_len(unsigned long, std::allocator<long long> const&) [46] long long* std::fill_n<long long*, unsigned long, long long>(long long*, unsigned long, long long const&)
[21] std::vector<long long, std::allocator<long long> >::size() const [36] std::vector<long long, std::allocator<long long> >::_M_default_initialize(unsigned long) [47] void std::_Destroy<long long*>(long long*, long long*) [22] void std::_Destroy_aux<true>::__destroy<long long*>(long long*, long long*) [19] std::vector<long long, std::allocator<long long> >::end() [48] void std::__fill_a<long long*, long long>(long long*, long long*, long long const&) [23] std::_Vector_base<long long, std::allocator<long long> >::_M_allocate(unsigned long) [18] std::vector<long long, std::allocator<long long> >::begin() [49] __gnu_cxx::__enable_if<std::__is_scalar<long long>::__value, void>::__type std::__fill_a1<long long*, long long>(long long*, long long*, long long const&)
[24] std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::_Vector_impl(std::allocator<long long> const&) [37] std::vector<long long, std::allocator<long long> >::vector(unsigned long, std::allocator<long long> const&) [50] operator new(unsigned long, void*)
[25] std::_Vector_base<long long, std::allocator<long long> >::_Vector_impl::~_Vector_impl() [38] std::vector<long long, std::allocator<long long> >::~vector()
[26] std::_Vector_base<long long, std::allocator<long long> >::_M_deallocate(long long*, unsigned long) [15] std::vector<long long, std::allocator<long long> >::operator[](unsigned long)